Hochschule Konstanz
Department of Electrical Engineering
and Information Technology

A Gaussian inverse Wishart PHD Filter using Stochastic Partitioning for Multiple Extended Object Tracking

Julian Böhler

Master Thesis

in the Study Programme

Master of Engineering
Electrical Systems Engineering (EIM)

at

Institute for System Dynamics (ISD),
HTWG Konstanz

Author:	Julian Böhler, B.Eng.
Enrolment Number:	296171
Day of submission:	15. April 2019
Processing Period:	October 2018 – April 2019
University of Applied Sciences:	HTWG Konstanz, Germany
Supervising Professor:	Prof. Dr.-Ing. Johannes Reuter
Supervisor:	Stefan Wirtensohn, M.Eng.

Abstract

This thesis deals with the object tracking problem of multiple extended objects. For instance, this tracking problem occurs when a car with sensors drives on the road and detects multiple other cars in front of it. When the setup between the senor and the other cars is in a such way that multiple measurements are created by each single car, the cars are called extended objects. This can occur in real world scenarios, mainly with the use of high resolution sensors in near field applications. Such a near field scenario leads a single object to occupy several resolution cells of the sensor so that multiple measurements are generated per scan. The measurements are additionally superimposed by the sensor's noise. Beside the object generated measurements, there occur false alarms, which are not caused by any object and sometimes in a sensor scan, single objects could be missed so that they not generate any measurements.

To handle these scenarios, object tracking filters are needed to process the sensor measurements in order to obtain a stable and accurate estimate of the objects in each sensor scan. In this thesis, the scope is to implement such a tracking filter that handles the extended objects, i.e. the filter estimates their positions and extents. In context of this, the topic of measurement partitioning occurs, which is a pre-processing of the measurement data. With the use of partitioning, the measurements that are likely generated by one object are put into one cluster, also called cell. Then, the obtained cells are processed by the tracking filter for the estimation process. The partitioning of measurement data is a crucial part for the performance of tracking filter because insufficient partitioning leads to bad tracking performance, i.e. inaccurate object estimates.

In this thesis, a Gaussian inverse Wishart Probability Hypothesis Density (GIW-PHD) filter was implemented to handle the multiple extended object tracking problem. Within this filter framework, the number of objects are modelled as Random Finite Sets (RFSs) and the objects' extent as random matrices (RM). The partitioning methods that are used to cluster the measurement data are existing ones as well as a new approach that is based on likelihood sampling methods. The applied classical heuristic methods are Distance Partitioning (DP) and Sub-Partitioning (SP), whereas the proposed likelihood-based approach is called Stochastic Partitioning (StP). The latter was developed in this thesis based on the Stochastic Optimisation approach by Granström et al. An implementation, including the StP method and its integration into the filter framework, is provided within this thesis.

The implementations, using the different partitioning methods, were tested on simulated random multi-object scenarios and in a fixed parallel tracking scenario using Monte Carlo methods. Further, a runtime analysis was done to provide an insight into the computational effort using the different partitioning methods. It emphasized, that the StP method outperforms the classical partitioning methods in scenarios, where the objects move spatially close. The filter using StP performs more stable and with more accurate estimates. However, this advantage is associated with a higher computational effort compared to the classical heuristic partitioning methods.

Acknowledgements

First of all, I give thanks to my Professor *Johannes Reuter*, who made it possible to write my Master's thesis in such an interesting and challenging field of research. Thank you for the support during the thesis, the time you have spent for the review process, the trust in me and finally, the freedom during the thesis to come up with own ideas and the possibility to implement them. I look forward to working further at the Institute of System dynamics and thank very much for this opportunity.

A further thank goes to *Stefan Wirtensohn* as my supervisor and discussion partner during the last months. He was always willing to give answers to my questions at all times and thus quickly brought light into the darkness. I am very thankful for the discussions, the answers, the suggestions and the review of my thesis.

I give another thank to *Tim Baur*, my fellow student during my Master's thesis. He inspired me many times and challenged my work in many discussions, which always was helpful.

The next expression of thanks goes to Sweden to *Karl Granström* for the electronic correspondence. I thank for his fast and kind answers to the questions that I have had to his papers. Due to his suggestion to have a glance at his current scientific work, I was inspired with new ideas for my thesis.

Last but not least I thank *Michael Schuster* for the excellent lectures in multi-sensor data fusion that I visited during my Master's course. The mediated fundamentals were challenging, but very worthwhile for the subject of the thesis. Without his lecture, I would have never gained that fast and profound insight into this topic.

Deposition

Hereby I declare that I wrote this thesis with the title

“A Gaussian inverse Wishart PHD Filter using Stochastic Partitioning
for Multiple Extended Object Tracking”

on my own and without inadmissible help. The ideas that I took over from other persons are denoted with its origins and further, I did not use any other sources than the cited ones.

Konstanz (Germany), 15. April 2019, __________________________

Contents

I. Nomenclature and symbols

General notations

a	scalar or constant a (small letter)
$\boldsymbol{a}$	vector $\boldsymbol{a}$ (small bold letter)
A	matrix A (capital letter)
$\lvert A \rvert$	determinant of matrix A
$\boldsymbol{A}$	set $\boldsymbol{A}$ (capital bold letter)
$\lvert \boldsymbol{A} \rvert$	cardinality of set $\boldsymbol{A}$
k	filter time step
t	sampling iteration step
$(\cdot)^{-1}$	inverse of $(\cdot)$
$(\cdot)^{T}$	transpose of $(\cdot)$
$(\cdot)!$	factorial of $(\cdot)$
$(\cdot)_{k}$	symbol $(\cdot)$ at time step k
$(\cdot)_{k\lvert k}$	corrected symbol $(\cdot)$ at time step k
$(\cdot)_{k\lvert k-1}$	predicted symbol $(\cdot)$ at time step $k-1$
$diag(\cdot)$	diagonal matrix of the input arguments $(\cdot)$
$\otimes$	Kronecker product

Frequently used specific notations

α	action of a sampling method
$\mathcal{A}$	data association
$\beta_{k\lvert k-1}(\cdot)$	spawning intensity
$\boldsymbol{B}_{k\lvert k-1}$	spawning RFS
$\boldsymbol{C}$	measurement cell in the context of SO
$\mathcal{C}$	components of the GIW-PHD filter
$d(\cdot)$	distance measure
d_{H}	Hellinger distance
d_{M}	Mahalanobis distance
$d_{\boldsymbol{W}}$	cell weight normalisation coefficient, absolute cell weight
$D_{k\lvert k}$	a posteriori PHD at time step k
$D_{k\lvert k-1}$	predicted PHD at time step $k-1$
$\delta_{i,j}$	Kronecker delta
η_{M}	update normalisation factor (multi-object case)
η_{S}	update normalisation factor (single-object case)
$f(\cdot)$	state transition function of a single object
$f[g]$	integral abbreviation: $\int f(x)g(x)dx$
F	linear transition matrix
$\gamma_{k}(\cdot)$	birth intensity
$\boldsymbol{\Gamma}_{k}$	birth RFS
$h(\cdot)$	measurement model function of a single object
H	linear measurement matrix
I_{d}	identity matrix of dimension d
$\mathbb{I}$	object index set
K	gain matrix
$\kappa_{k}(\cdot)$	clutter intensity
$\Lambda_{i,j}$	abbreviation for a distance measure between measurements $\boldsymbol{z}^{(i)}$ and $\boldsymbol{z}^{(j)}$

$\mathcal{L}^{\wp}$	relative likelihood of partition $\wp$
$\mathbb{M}$	measurement index set
ν	inverse Wishart degrees of freedom
$\omega_{\wp}$	partition normalisation coefficient, relative partition weight
p_D	detection probability
$p(\cdot)$	probability distribution of $(\cdot)$
$p(\cdot \mid \cdot)$	conditional probability distribution
$\wp^{(t)}$	partition at iteration t
$\wp_{\alpha}^{(t)}$	partition at iteration t after applying an action α
$\wp(\cdot)$	corresponding partition to a given assignment $(\cdot)$
$P_{(\cdot)}$	process covariance matrix
$\mathcal{P}$	partition set
$\xrightarrow{P}$	prediction step
$\varphi_m^{(t)}$	element m of the assignment vector at iteration t
$\boldsymbol{\varphi}^{(t)}$	assignment vector at iteration t
$\boldsymbol{\varphi}_{\alpha}^{(t)}$	assignment vector at iteration t after applying an action α
S_k	innovation covariance matrix
$\mathcal{S}$	surveillance volume
Σ	scalar acceleration standard derivation
τ	decay constant
T_S	sampling time
θ	manoeuvre correlation time
$\xrightarrow{U}$	update/correction step
V	inverse Wishart scale matrix
$\boldsymbol{W}$	measurement cell in context of StP and partitions
$\boldsymbol{W}_c^{(t)}$	cell of StP iteration t with identifier c
$\boldsymbol{W}_c^{(\wp)}$	cell of pth partition and cell identifier c
$\mathcal{W}_{\wp}$	(absolute) partition weight
$\mathcal{W}_{\wp}^{(t)}$	partition weight at StP iteration t
$\mathcal{W}_{\wp}^{(\alpha)}$	partition weight at StP iteration t after applying action α
$\boldsymbol{x}_k$	kinematics state vector
$\hat{\boldsymbol{x}}$	kinematics state estimate
ξ_k	extended object state vector $(\boldsymbol{x}_k, X_k)$ at time step k
X_k	extension matrix
$\boldsymbol{X}_k$	finite set of object states
$\hat{\boldsymbol{X}}_k$	set of estimated objects
$\boldsymbol{X}^k$	sequence of state sets up to and including time step k
$\mathcal{X}$	hyperspace of a RFS
$\mathcal{X}_0$	underlying space of $\mathcal{X}$
$\boldsymbol{z}_k$	measurement at time step k
$\hat{\boldsymbol{z}}_{k\|k-1}$	predicted measurement
$\boldsymbol{Z}_k$	finite set of measurements
$\boldsymbol{Z}^k$	sequence of measurement sets up to and including time step k

1. Preamble

With the first chapter, an introduction to this thesis is outlined starting in section 1.1 with a brief description of the problems related to the tracking of multiple objects with a perception system. Additionally, the approaches investigated for tracking problems at the Institute of System Dynamics are mentioned. In section 1.2, the fundamental scientific works that are incorporated in this thesis, are revealed. It is described, which approach is used to treat the multi-object tracking problem. In section 1.3, the structure of the thesis is given with a summarizing description of the single chapters' content.

1.1. Introduction

In current years, the endeavour of autonomy for vehicles undergoes huge advances. In this context, the environment perception and detection using sensor systems are of outstanding importance. Modern sensors based on infrared, laser, radar, sonar or camera technology are used to track moving objects in the environment. Tracking, as defined in [1, p. 2], is the processing of sensor measurements that are obtained from an object to maintain an estimate of its current state. A state of an object is comprised of kinematic components like position, velocity and acceleration or any other components or object features. It is estimated through algorithms that process the sensor measurements. With regard to a multi-object scenario, where a large but unknown number of objects is moving in a surveillance area, a sensor perception system will create a large number of object originated measurements that are noise corrupted. Additionally, there will be false alarms (clutter) that further increase the number of collected sensor measurements. The challenges of such a scenario a varied. Starting with the fact, that the number of objects is unknown. Then, it is obscure which measurements are caused by which object, if all objects caused any measurement at all and whether several measurements are clutter or not. That makes tracking of all objects difficult and creates the need for tracking algorithms that solve these problems.

In general, it can be distinguished between scenarios where the object and sensor setup creates at most one measurement per time step or multiple measurements per time step. In the former case, the tracked objects are called point objects and in the latter case, they are called extended objects. This thesis emphasis scenarios, where multiple extended objects are tracked. That means, the estimated object state consists of a kinematic state in combination with an extension state. In recent years a large number of algorithmic approaches have been proposed for this purpose. At the Institute of System Dynamics (ISD) at the HTWG Konstanz different approaches to multiple extended object tracking are investigated. Schuster [2] developed a method, the Multi Detection Joint Integrated Probabilistic Data Association (MD-JIPDA), that goes beyond the state of the art. This approach is based on the classical Bayesian filter with a clever strategy to reduce the possible allocation hypotheses. On the other hand, random finite set methods [3] [4], have increasingly become the focus of attention in recent years, not only for scientific considerations but also for practical applications. Baur [5] developed a method that combines the PHD filter as multi-object tracker with B-splines for extension estimation when using a measurement model that is comparable with the characteristics of LIDAR measurements. This thesis deals with the method of Granström [6], called the Gaussian inverse Wishart Probability Hypothesis Density (GIW-PHD) filter and new approaches of measurement partitioning.

1.2. Scope of this work

In this thesis, the multiple extended object tracker, called GIW-PHD filter, is implemented and tested in a simulation environment with multiple objects that have elliptical extent. The applied filter uses the random matrix framework of Koch [7] for the extent estimation and combines it with the extended object PHD filter framework [8]. The crucial part of this tracking algorithm is the partitioning of measurement data into cells containing measurements that belong together, i.e. they originate from the same object. For this problem heuristic methods are used traditionally that are usually generate cells based on the distance between the measurements. Common methods are e.g. Density-Based Spatial Clustering of Applications with Noise (DBCSAN), its advancement Ordering Points To Identify the Clustering Structure (OPTICS) or Distance Partitioning, that was introduced in [9] and is used in [6]. These methods have in common that on the one hand, they work well when objects are separated with a certain distance, but on the other hand, they lead to a poor filter performance when the objects are closely spaced.

Due to the mentioned drawbacks, a promising new approach of the recent research called Stochastic Optimisation (SO), which was presented in [10] for a Poisson Multi-Bernoulli Mixture (PMBM) Filter, is taken into account for improving the filter performance with closely spaced objects. The aim of this approach is to find high likely data associations by doing random actions that change the initial data association. The actions are inspired by the two stochastic sampling methods called Gibbs Sampling and Metropolis Hastings. The approach of SO is not applicable to the GIW-PHD framework directly and therefore is adapted and integrated into the GIW-PHD filter in this thesis. The resulting method is called Stochastic Partitioning (StP) and to the author's knowledge neither exist nor has been published yet.

To proof the applicability and performance of Stochastic Partitioning with the GIW-PHD filter, it is analysed using sequential Monte Carlo methods in tracking scenarios with multiple objects. The results are compared with the heuristic partitioning methods Distance Partitioning and Sub-Partitioning using multi-object miss distances and doing a runtime comparison.

1.3. Structure of thesis

The remainder of this thesis is structured as follows. In chapter 2, the fundamentals of object tracking are given with the introduction of the state estimation problem, motion and measurement modelling, the Bayesian filter framework, the derivation of the linear Kalman filter and finally a short glance at non-linear Kalman filters. In chapter 3, the multi-object tracking problem is described with the definition of its single assumptions, the classical data association approaches to solve this problem, the novel random finite set approach [11] and the motion and measurement models for multi-object states. In the end of the chapter, the PHD filter for point objects of Vo et al. [12] is presented. Chapter 4 treats the tracking problem of a single extended objects with a review of the measurement model according to Gilholm et al. [13] and the random matrix approach of Koch [7]. In chapter 5, the GIW-PHD filter, as multiple extended object tracker [6], is presented. This filter is investigated in this thesis for the advancement of a likelihood-based partitioning method. Classical methods for partitioning and likelihood-based approaches are presented in chapter 6. Further, it contains the presentation of the new proposed Stochastic Partitioning method for the GIW-PHD filter that is based on the work of Granström in [14]. Chapter 7 describes the integration of the proposed Stochastic Partitioning method into the GIW-PHD filter framework. Finally in chapter 8, the proposed implementation with Stochastic Partitioning is evaluated with random multi-object scenarios and Monte Carlo simulations of a defined tracking scenario. For the performance evaluation, implementations with the classical partitioning methods Distance Partitioning and Sub-Partitioning are compared with the implementation using Stochastic Partitioning. For this, the multi-object metrics OSPA and H-OSPA, the latter incorporates the Hellinger distance, are used. Further, the runtime of the single partitioning methods is compared. In chapter 9, the achieved work in this thesis is summarised and a glance at possible future work is given.

2. Fundamentals of object tracking

This chapter introduces the fundamentals of object tracking and starts in section 2.1 with the definition of the state estimation problem and the related basic notation of state, measurements and their set notations. In section 2.2, the general motion and measurement model of a point object is given with some assumptions that lead to their linear model versions. A short derivation of the probabilistic Bayesian filtering framework is given in section 2.3, which is a powerful tool to process measurement data recursively in order to create a state estimate using probability densities. Due to the fact that the Bayesian filter framework is of theoretical nature, for practical scenarios Gaussian distributions can be assumed. With these assumptions, the well-known linear Kalman filter (LKF) can be derived as depicted in section 2.4. Other nonlinear solutions are mentioned in section 2.5.

2.1. The state estimation problem

The main purpose of object tracking is to create an estimate of one or more object states $\boldsymbol{x}$. In general, it is also possible to estimate constant parameters $\boldsymbol{\theta}$. Both cases have in common that in real world problems the state or parameter is either not directly measureable or the measurements are error-prone. Estimation can be considered in discrete as well as continuous time, see [15]. The following depiction is bounded for discrete time steps. And for clarity, the state estimation problem of a single point object is assumed for the given depiction. A point object is an object that gives rise of at most one measurement per time step. This is true for applications where the object extent is negligible in comparison to the sensor resolution, mostly in long range applications such as the classical radar airplane tracking [1]. For the multi-object state estimation problem see chapter 3.
Let the true state at time t_k, i.e. at discrete time step k, be $\boldsymbol{x}_k \epsilon \mathbb{R}^{n_x}$. The state is a vector that consists of valued variables, given as

$$\boldsymbol{x}_k = \left(x^{(1)}, \dots, x^{(n_x)}\right)_k^T. \tag{2.1}$$

Further, let $\boldsymbol{X}^k$ be a set that contains all true states up to and including to time t_k, denoted as

$$\boldsymbol{X}^k = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \dots, \boldsymbol{x}_k,\}. \tag{2.2}$$

The true state $\boldsymbol{x}_k$ is observable by measurements $\boldsymbol{z}_k \epsilon \mathbb{R}^{n_z}$ that are given by

$$\boldsymbol{z}_k = \left(z^{(1)}, \dots, z^{(n_z)}\right)_k^T. \tag{2.3}$$

The set of all measurements up to and including to time step k is defined as

$$\boldsymbol{Z}^k = \{\boldsymbol{z}_0, \boldsymbol{z}_1, \dots, \boldsymbol{z}_k,\}. \tag{2.4}$$

The estimated state $\hat{\boldsymbol{x}}$ at time step k given all measurement up to and including time step l is denoted as $\hat{\boldsymbol{x}}_{k|l}$. Now it can be distinguished between three cases [15]. First, when time $t_l > t_k$ with $l > k$, the estimation problem is called smoothing. That means, the estimation in the past at time t_k is enhanced by using information at time t_l. Second, when time $t_l < t_k$ with $l < k$, the estimation problem is called prediction, whereby information at time t_l is used to predict the estimate in the future at time t_k. Finally, the estimation problem that is known as filtering, as described in this section, occurs when time $t_l = t_k$ with $t = l$.

2.2. Motion and measurement modelling

To solve the filtering problem, it is necessary to model the dynamic motion of the observed object and the relationship of the measurements and the object.

For the motion model this can be done in general using a nonlinear function that is dependent on the object state $\boldsymbol{x}_k$, the constant parameters $\boldsymbol{\theta}$, a random noise $\boldsymbol{w}_k$ and the system input $\boldsymbol{u}_k$ [2]:

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}, \boldsymbol{\theta}, \boldsymbol{w}_{k-1}) \tag{2.5}$$

For (2.5) a few assumptions can be made in the context of object tracking as noted in [15, p. 38] and [2, p. 10]. The state estimation often assumes the parameters $\boldsymbol{\theta}$ as known, otherwise they are included in the state vector. Further, the tracked object is not cooperative and therefore does not transmit any information about the system input $\boldsymbol{u}_k$ to the surveillance system. This leads to an unknown system input $\boldsymbol{u}_k$ that is cancelled in (2.5) but has to be taken into account via the random noise. Thus, the noise term $\boldsymbol{w}_k$ itself is assumed to be additive with expectation value zero. The depicted assumptions above lead to a simplified motion model given as

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}) + \boldsymbol{w}_{k-1}. \tag{2.6}$$

The relationship between measurements $\boldsymbol{z}_k$ and object state $\boldsymbol{x}_k$ can be modelled in the form

$$\boldsymbol{z}_k = h(\boldsymbol{x}_k, \boldsymbol{\theta}, \boldsymbol{e}_k). \tag{2.7}$$

Here $\boldsymbol{e}_k$ describes the random measurement noise term and with the same assumptions of the motion modelling the measurement model can be written simplified as

$$\boldsymbol{z}_k = h(\boldsymbol{x}_k) + \boldsymbol{e}_k. \tag{2.8}$$

The discrete time step is related as

$$t_k = t_{k-1} + T_s(k) \tag{2.9}$$

where $T_s(k)$ is the sampling time of the sensor at time step k. In general, the sampling time can vary over time, but in this thesis a constant sampling time $T_s(k) = T_s$ is assumed.
The motion and measurement models that are used in this thesis are described in chapter 4. A comprehensive survey of mathematical motion models of manoeuvring objects for 2D and 3D observation spaces can be found in [16]. For a survey on measurement models see [17]. The linear motion and measurement model are introduced in section 2.4.

2.3. Bayesian Filtering

A well-known method to create a state estimate out of received measurement data is the probabilistic Bayesian framework. Due to the fact that the object motion process and the measurement process are noisy processes with uncertainties one uses probability distributions to describe them. Defining a probability $p(\boldsymbol{X}^k|\boldsymbol{Z}^k)$ directly is difficult and for that reason one exploits Bayes' theorem denoted as

$$p(\boldsymbol{X}^k|\boldsymbol{Z}^k) = \frac{p(\boldsymbol{Z}^k|\boldsymbol{X}^k)p(\boldsymbol{X}^k)}{p(\boldsymbol{Z}^k)} \tag{2.10}$$

Now the aim is to find a recursive solution that creates an estimation for a single object state $p(\boldsymbol{x}_k|\boldsymbol{Z}^k)$ at time step k for received measurement data $\boldsymbol{Z}^k$. For that (2.10) needs to be rearranged and all probabilities on the right side of the equation are split as

$$p(\boldsymbol{Z}^k|\boldsymbol{X}^k) = p(\boldsymbol{z}_k, \boldsymbol{Z}^{k-1}|\boldsymbol{X}^k) = p(\boldsymbol{z}_k|\boldsymbol{Z}^{k-1}, \boldsymbol{X}^k)p(\boldsymbol{Z}^{k-1}|\boldsymbol{X}^k), \tag{2.11}$$

$$p(\boldsymbol{X}^k) = p(\boldsymbol{x}_k, \boldsymbol{X}^{k-1}) = p(\boldsymbol{x}_k|\boldsymbol{X}^{k-1})p(\boldsymbol{X}^{k-1}), \tag{2.12}$$

and

$$p\big(\boldsymbol{Z}^k\big) = p\big(\boldsymbol{z}_k, \boldsymbol{Z}^{k-1}\big) = p\big(\boldsymbol{z}_k\big|\boldsymbol{Z}^{k-1}\big)p\big(\boldsymbol{Z}^{k-1}\big). \tag{2.13}$$

For $p\big(\boldsymbol{Z}^{k-1}\big|\boldsymbol{X}^k\big)$ in (2.11) the principle of causality leads to

$$p\big(\boldsymbol{Z}^{k-1}\big|\boldsymbol{X}^k\big) = p\big(\boldsymbol{Z}^{k-1}\big|\boldsymbol{X}^{k-1}\big), \tag{2.14}$$

which implies that the measurement data $\boldsymbol{Z}^{k-1}$ at time step $k-1$ only does depend on the state $\boldsymbol{X}^{k-1}$ at the same time step. In other words, the state at time step k does not influence the measurement data in the past.

When inserting (2.11) with (2.14), (2.12) and (2.13) into (2.10), a recursive formula follows as

$$p\big(\boldsymbol{X}^k\big|\boldsymbol{Z}^k\big) = \frac{p(\boldsymbol{z}_k|\boldsymbol{Z}^{k-1}, \boldsymbol{X}^k)p\big(\boldsymbol{x}_k\big|\boldsymbol{X}^{k-1}\big)}{p\big(\boldsymbol{z}_k\big|\boldsymbol{Z}^{k-1}\big)} \frac{p\big(\boldsymbol{Z}^{k-1}\big|\boldsymbol{X}^{k-1}\big)p\big(\boldsymbol{X}^{k-1}\big)}{p\big(\boldsymbol{Z}^{k-1}\big)}, \tag{2.15}$$

where the second quotient corresponds to (2.10) at time step $k-1$. Thus it follows that

$$p\big(\boldsymbol{X}^k\big|\boldsymbol{Z}^k\big) = \frac{p(\boldsymbol{z}_k|\boldsymbol{Z}^{k-1}, \boldsymbol{X}^k)p\big(\boldsymbol{x}_k\big|\boldsymbol{X}^{k-1}\big)}{p\big(\boldsymbol{z}_k\big|\boldsymbol{Z}^{k-1}\big)} p\big(\boldsymbol{X}^{k-1}\big|\boldsymbol{Z}^{k-1}\big). \tag{2.16}$$

To go further, two assumptions are assumed that are common for many real systems but must not be assumed as valid in the general case. The first assumption is that the state transition density $p\big(\boldsymbol{x}_k\big|\boldsymbol{X}^{k-1}\big)$ is assumed to be a Markov process, which means the state at time step k is only dependent on the state at time step $k-1$ and is independent of all other previous states. This leads to $p\big(\boldsymbol{x}_k\big|\boldsymbol{X}^{k-1}\big) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$. With this assumption it is possible to specify a transition density for the model as shown in (2.5). The second assumption has an effect on the measurement density $p(\boldsymbol{z}_k|\boldsymbol{Z}^{k-1}, \boldsymbol{X}^k)$ and claims that the measurement at time step k is on the one hand independent of preceding measurements $\boldsymbol{Z}^{k-1}$ and on the other hand depends only on the state $\boldsymbol{X}^k$. Thus, the density $p(\boldsymbol{z}_k|\boldsymbol{Z}^{k-1}, \boldsymbol{X}^k)$ simplifies to $p(\boldsymbol{z}_k|\boldsymbol{x}_k)$. Likewise, with this assumption it is possible to specify a measurement density $p(\boldsymbol{z}_k|\boldsymbol{x}_k)$ for the measurement model given in (2.7).

Both assumptions further simplify (2.16) to the recursive Bayes filter given as

$$p\big(\boldsymbol{X}^k\big|\boldsymbol{Z}^k\big) = \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})}{p\big(\boldsymbol{z}_k\big|\boldsymbol{Z}^{k-1}\big)} p\big(\boldsymbol{X}^{k-1}\big|\boldsymbol{Z}^{k-1}\big). \tag{2.17}$$

As mentioned in the beginning of the section, only the current single object state $\boldsymbol{x}_k$ is of interest and not in the previous ones, i.e. one needs to calculate the density $p\big(\boldsymbol{x}_k\big|\boldsymbol{Z}^k\big)$. To achieve this, the marginal distribution of (2.17) has to be calculated by integrating over all previous states $\boldsymbol{x}_0 \ldots \boldsymbol{x}_{k-1}$ leading to the conditional a posteriori probability of the state:

$$p\big(\boldsymbol{x}_k\big|\boldsymbol{Z}^k\big) = \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k)}{p\big(\boldsymbol{z}_k\big|\boldsymbol{Z}^{k-1}\big)} \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})p\big(\boldsymbol{x}_{k-1}\big|\boldsymbol{Z}^{k-1}\big)\, d\boldsymbol{x}_{k-1}. \tag{2.18}$$

The Bayes filtering algorithm is obtained by splitting (2.18) into the *prediction step,* also known as *Chapman-Kolmogorov equation*, stated as

$$p\big(\boldsymbol{x}_k\big|\boldsymbol{Z}^{k-1}\big) = \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})p\big(\boldsymbol{x}_{k-1}\big|\boldsymbol{Z}^{k-1}\big)\, d\boldsymbol{x}_{k-1} \tag{2.19}$$

and the *correction-* or *update-step* given as

$$p\big(\boldsymbol{x}_k\big|\boldsymbol{Z}^k\big) = \eta_S\, p(\boldsymbol{z}_k|\boldsymbol{x}_k)\, p\big(\boldsymbol{x}_k\big|\boldsymbol{Z}^{k-1}\big), \tag{2.20}$$

where the normalization factor $\eta_S = \frac{1}{p(\boldsymbol{z}_k|\boldsymbol{Z}^{k-1})}$ has to be calculated using the law of total probability by integrating over all states that could have caused the given measurement data:

$$p(\boldsymbol{z}_k|\boldsymbol{Z}^{k-1}) = \int p(\boldsymbol{z}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{Z}^{k-1})\, d\boldsymbol{x}_k. \tag{2.21}$$

With the Bayes filter a tool is provided that allows to process measurement data recursively using the filter's prediction step (P) and update step (U) as illustrated as follows:

$$\cdots \overset{U}{\rightarrow} p(\boldsymbol{x}_{k-1}|\boldsymbol{Z}^{k-1}) \overset{P}{\rightarrow} p(\boldsymbol{x}_k|\boldsymbol{Z}^{k-1}) \overset{U}{\rightarrow} p(\boldsymbol{x}_k|\boldsymbol{Z}^{k}) \overset{P}{\rightarrow} \cdots \tag{2.22}$$

To extract the information of interest (position, velocity,…) out of the posterior probability $p(\boldsymbol{x}_k|\boldsymbol{Z}^k)$, in the single state case a Bayes optimal state estimator can be used [18]. On the one hand this can be the maximum a posteriori (MAP) estimator given as

$$\hat{\boldsymbol{x}}_{k|k}^{MAP} = \arg \max_{\boldsymbol{x}_k \in \mathbb{R}^{n_x}} p(\boldsymbol{x}_k|\boldsymbol{Z}^k). \tag{2.23}$$

On the other hand, the expected a posteriori (EAP) estimation can be applied, defined as

$$\hat{\boldsymbol{x}}_{k|k}^{EAP} = \int \boldsymbol{x}_k \; p(\boldsymbol{x}_k|\boldsymbol{Z}^k) d\boldsymbol{x}_k. \tag{2.24}$$

Within the Bayes filter framework, it is desired that the propagated density of the state $\boldsymbol{x}$, cf. (2.22), has the same functional form. That means, the densities $p(\boldsymbol{x}_k|\boldsymbol{Z}^{k-1})$ and $p(\boldsymbol{x}_k|\boldsymbol{Z}^k)$ are in the same probability distribution family for all time steps k. This property then is called conjugacy. For an initial measurement density $p(\boldsymbol{z}_k|\boldsymbol{x}_k)$ the prior density that gives the same posterior according to the probability distribution family is called conjugate prior.

This concept is important to apply the Bayes filter algorithm in real world applications. Assuming the initial density $p(\boldsymbol{x}_{k-1}|\boldsymbol{Z}^{k-1})$ to be a Gaussian as well as transition density (2.19) and the measurement density (2.20), the posterior density also will be a Gaussian of the same distribution family. These assumptions lead to the derivation of the well-known linear Kalman filter, which is presented in the next section.

2.4. Linear Kalman filter

The linear Kalman filter (LKF) was developed in 1960 and was presented by Rudolf E. Kalman in [19]. Today it is widely used and listed in a lot of textbooks as in [1] [20] [21].For the derivation of the linear Kalman filter three assumptions are made [20]:

A.1. The motion model and measurement model in (2.5) and (2.7) are linear and follow:

$$\boldsymbol{x}_k = F\boldsymbol{x}_{k-1} + \boldsymbol{w}_{k-1}, \tag{2.25}$$

$$\boldsymbol{z}_k = H\boldsymbol{x}_k + \boldsymbol{e}_k, \tag{2.26}$$

where F is the constant transition matrix and H the constant measurement matrix.

A.2. $\boldsymbol{w}_{k-1}$ and $\boldsymbol{e}_k$ are the respective uncorrelated and Gaussian noise sequences with zero mean and covariance matrices Q_{k-1} and R_k.

A.3. The posterior probability density of the object state $p(\boldsymbol{x}_{k-1}|\boldsymbol{Z}^{k-1})$ at time step $k-1$ can be approximated with a Gaussian with mean $\hat{\boldsymbol{x}}_{k-1|k-1}$ and covariance $P_{k-1|k-1}$[1]. This assumption can be stated as $p(\boldsymbol{x}_{k-1}|\boldsymbol{Z}^{k-1}) \cong p(\boldsymbol{x}_{k-1}|\hat{\boldsymbol{x}}_{k-1|k-1}, P_{k-1|k-1})$.

[1] Note that the 3rd assumption implies a negligence of higher moments besides the mean (1st moment) and covariance matrix (2nd moment).

With assumptions A.1 and A.2 the probability distributions for the transition density and measurement density can be defined. The transition density can be written as normal distribution given as

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) = p(\boldsymbol{w}_k) = p(\boldsymbol{x}_k - F\boldsymbol{x}_{k-1}) = \mathcal{N}(\boldsymbol{x}_k; F\boldsymbol{x}_{k-1}, Q_{k-1}), \tag{2.27}$$

as well as the measurement density respectively as

$$p(\boldsymbol{z}_k|\boldsymbol{x}_k) = p(\boldsymbol{e}_k) = p(\boldsymbol{z}_k - H\boldsymbol{x}_k) = \mathcal{N}(\boldsymbol{z}_k; H\boldsymbol{x}_k, R_k). \tag{2.28}$$

Further, assumption A.3 leads to

$$p(\boldsymbol{x}_{k-1}|\mathbf{Z}^{k-1}) = \mathcal{N}(\boldsymbol{x}_{k-1}; \hat{\boldsymbol{x}}_{k-1|k-1}, P_{k-1|k-1}). \tag{2.29}$$

With these three normal distributions the Kalman filter can be derived by inserting them in the Chapman-Kolmogorov equation (2.19), the update step (2.20) and the calculation of the normalization coefficient (2.21). It can be seen that for the derivation an integral over the product of two Gaussian distributions has to be calculated, which can be solved using the Gaussian product theorem. In Table 1 the resulting filter equations are presented. For a whole derivation see e.g. [20, pp. 25-30].

Table 1: The linear Kalman filter algorithm

Prediction of the estimated state and covariance matrix:

1: $\hat{\boldsymbol{x}}_{k|k-1} = F\hat{\boldsymbol{x}}_{k-1|k-1}$

2: $P_{k|k-1} = FP_{k-1|k-1}F^T + Q_{k-1}$

Computation of the predicted measurement, innovation covariance matrix and Kalman gain:

3: $\hat{\boldsymbol{z}}_{k|k-1} = H\boldsymbol{x}_{k|k-1}$

4: $S_k = HP_{k|k-1}H^T + R_k$

5: $K_k = P_{k|k-1}H^TS_k^{-1}$

Compute the posterior state estimation and covariance matrix

6: $\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + K_k(\boldsymbol{z}_k - \hat{\boldsymbol{z}}_{k|k-1})$

7: $P_{k|k} = P_{k|k-1} - K_kHP_{k|k-1}$

2.5. Nonlinear Kalman and particle filtering

The linear Kalman filter presented in the section above is applicable when all of the mentioned three assumptions hold. Now one can suggest scenarios where these assumptions do not hold, e.g. when the linear dynamic and measurement equations in (2.25) and (2.26) do not model the real scenario accurately enough. As mentioned in section 2.2, the dynamic and measurement models also can be nonlinear functions. This is the case when e.g. measurement data is received in range and azimuth and not directly in Cartesian measurements. Further, the motion model could involve the steering angles, so that additional nonlinearities exist. These conditions require new approaches to obtain an accurate tracking performance.

One possibility is the so called extended Kalman filter (EKF). This filter can be derived under the assumptions A.2 and A.3 in section 2.2, but with nonlinear motion and measurement models [20, pp. 31-36]. The key idea behind the EKF is to linearize all nonlinear models so that the traditional LKF shown in Table 1 can be applied.

Another possibility is to use the Unscented Kalman filter (UKF) which was presented by Julier and Uhlman in [22]. The key idea there is to avoid the linearization of the EKF, and instead use the so called unscented transform. This provides a set of sigma points that parametrize the means and covariance matrices of the probability distributions. The derivation of the UKF can be found in [20, pp. 26-42].

A detailed discussion of non-linear Kalman filter variants, which also includes variants that are commonly less known as the central difference filter, the first order divided difference filter and the iterated extended Kalman filter, is given in [23].

Apart from the EKF and UKF, which can be seen as analytic approximations to the derived Bayes filter in section 2.3, another possibility is to approximate the distributions in the Bayes filter in a discrete way. This idea is found in literature as particle filters [20, p. 46] and their key idea is to approximate the distributions with random samples called particles. A high number of particles at a certain point represents a high probability, whereas less particles determine a low probability.

The EKF, UKF and the particle filter are just mentioned shortly for completeness, but are not further discussed because only LKF equations are used in this thesis for the linear motion and measurement model in the GIW-PHD filter.

3. Multi-object tracking

In the previous chapter, the state estimation problem was introduced namely for a single point object. For the LKF it was assumed that the measurements originate from one object without the presence of clutter. In the context of object tracking, one has to extend the problem definition because an object could create no detection at a time step or received measurements could be false alarms. These problems are addressed for multiple point objects in this chapter, which leads to the formulation of the multi-object tracking problem. In section 3.1, the assumptions for the multi-object tracking problem are listed. Further, the notation of the set values of multi-object state and measurements are introduced. Section 3.2 briefly addresses the classical data association approaches to solve the multi-object tracking problem, whereas section 3.3 introduces the Random Finite Set (RFS) approach, which is subject of current research in the field of multi-object tracking. Within this section, the finite set statistics (FISST) are addressed as well as the formulation of the multi-object Bayes filter with the approximation as probability density function. In section 3.4, the multi-object state transition modelling and measurement modelling is discussed, which is needed for presentation of the Gaussian mixture PHD filter in section 3.5.

3.1. The multi-object tracking problem

In classical single object tracking problems, e.g. tracking an airplane with radar sensors [1], the setup between object and sensor leads the airplane to occupy only one resolution cell of the radar sensor. This means an object creates at most one measurement per time step or when it is not detected no measurement. Due to this the object can be assumed as so called point objects. Now for the definition of the multi-object tracking problem all objects that are existent are assumed to be point objects and the following assumptions are supposed:

A.1. Each measurement that is caused by an object is noise corrupted.

A.2. The detection probability of each object is less than one and therefore it is not known whether an object has created a measurement at the current time step or not.

A.3. Beside object originated measurements there exist false alarms or clutter measurements.

A.4. The origin of each measurement is unknown, i.e. one does not know whether one measurement is clutter or object-generated.

A.5. The number of existing objects is unknown.

Let $n_{\boldsymbol{X}_k}$ be the unknown number of objects that exist at time step k. $\boldsymbol{x}_k^{(i)}$ denotes the ith object state at time step k and thus the set of all present objects is given as

$$\boldsymbol{X}_k = \left\{\boldsymbol{x}_k^{(i)}\right\}_{i=1}^{n_{X_k}}. \tag{3.1}$$

Let the number of object originated measurements and clutter be $n_{\boldsymbol{Z}_k}$ and $\boldsymbol{z}_k^{(j)}$ the jth received measurement of the sensor at time step k. The set of measurements can be written as

$$\boldsymbol{Z}_k = \left\{\boldsymbol{z}_k^{(j)}\right\}_{j=1}^{n_{Z_k}}. \tag{3.2}$$

Further, let all received measurements up to and including to time step k be $\boldsymbol{Z}^k = \{\boldsymbol{Z}_m\}_{m=1}^k$ as already defined in (2.4). The difference of the set here is, that it contains measurements of multiple objects and clutter, whereas the set $\boldsymbol{Z}^k$ in (2.4) only contains measurements of a single object without clutter.

Now the multi-object tracking problem can be seen as an extension of the single object tracking problem and is a joint estimation problem. Multiple measurements obtained from multiple objects have to be processed to obtain both the objects' current states and the number of present objects [15]. I.e. at every time step k the purpose is to estimate $\boldsymbol{X}_k$ given the measurements $\boldsymbol{Z}^k$, which can be expressed in a probabilistic way as the density

$$p(\boldsymbol{X}_k|\boldsymbol{Z}^k). \tag{3.3}$$

To solve the multi-object tracking problem, a classical way is to associate single measurements to single objects, so that every object can be processed separately using a single object tracker. The data association is addressed briefly in the next section to give an idea of the basic opportunities.

3.2. Data association approaches

Data association means to assign each measurement either to an object, that has potentially created the measurement, or to clutter. This is a crucial part of multi-object tracking algorithms because false assignments can lead from poor tracking results up to the divergence of a tracking algorithm. Approaches for solving the data association problem can be found in many textbooks, e.g. [1].

3.2.1. Global nearest neighbour

In the single object case the data association problem consists of the decision whether the object exists or not and in case, what measurement should be assigned to it. That means, the problem can be solved locally with the simple nearest neighbour (NN) approach, which implies that the measurement closest to the object estimate is the most likely one that has been originated from the true object. I.e. that measurement that minimizes the distance

$$d_M^2 = \left(\boldsymbol{z}_k^{(i)} - \hat{\boldsymbol{z}}_{k|k-1}\right)^T S_k^{-1} \left(\boldsymbol{z}_k^{(i)} - \hat{\boldsymbol{z}}_{k|k-1}\right) \tag{3.4}$$

is associated to the predicted object. S_k^{-1} and $\hat{\boldsymbol{z}}_{k|k-1}$ were defined in the LKF algorithm in Table 1. $\boldsymbol{z}_k^{(i)}$ is the ith measurement. The distance d_M^2 is called squared Mahalanobis distance and also can be used to exclude measurements around an estimate, which is called gating [2].

For the multi-object case the data association problem is becoming more complex because local methods like NN would perform insufficient. One has to do a global assignment that involves all measurement to object assignments jointly. This can be done with the global nearest neighbour (GNN) data association, that minimizes a total cost, e.g. a cost function that contains the sum of all squared distances. The minimization of such a cost function is an optimisation problem, which can be solved with different algorithms like Auction algorithm [24] or Munkres algorithm [25]. After the assignment, i.e. every measurement is assigned to an object estimate, each pair can be filtered by using a single object Bayes filter.

Whether in single- or multi-object case the NN and GNN approach make hard decisions subject to the assignment of a measurement to an object estimate. This can lead in complex scenarios to insufficient results, therefore in the following section a probabilistic approach, that is a soft version of GNN is addressed briefly.

3.2.2. Joint Probabilistic Data Association

In the single object case the soft version of NN is called probabilistic data association (PDA) [26]. Analogously the soft version of GNN is joint probabilistic data association (JPDA) when a known number of multiple objects exists [15]. The key idea behind the PDA approach is the consideration of each possible measurement association to an estimate. For each valid measurement, that falls inside a gate of an object estimate, a measurement update is performed and the corresponding likelihood that the selected measurement is object originated, is computed. The a posteriori state estimation then is obtained by the weighted sum of the updated state estimations [2]. In a multi-object case the JPDA approach as an extension of the PDA is applied. To avoid a conflicting measurement to object estimation assignment due to the presence of multiple objects, the JPDA uses joint association events and joint association probabilities [27]. Further details are given in [1] [20]. The JPDA approach is applicable only when the tracking scenario has a fixed and known object number [27]. To handle an unknown and time varying object number the joint integrated PDA approach has been developed [28] [29].

3.2.3. Multi hypothesis tracking

Multi hypothesis tracking (MHT) is a deferred decision approach to handle the data association problem [30]. As the name itself implies, the MHT approach processes a set of association hypotheses in every time step. In every time step for each given object estimation an update with every received measurement is performed resulting in a new set of association hypothesis. Then, each hypothesis in the new set is updated with every new received measurement and so on. Due to this, the number of hypothesis increases very fast and leads after a short time to computationally intractability. To stem this, the pruning of hypotheses with a low posterior probability, and the merging of quite similar hypotheses, is necessary.

The three approaches in section 3.2.1, 3.2.2 and 3.2.3 of handling the data association problem have in common, loosely speaking, that they first do a measurement to estimate assignment and then update individually every single measurement estimate pair. So they can be seen as an extension of the corresponding single object tracker version. A more closed approach that handles respectively all objects and all measurements as a unit and avoids the explicit data association is the Random Finite Set approach presented in next section. This approach has been established in the last decades and is currently of high interest in the research field of data fusion.

3.3. Random finite set approaches

The Random Finite Set (RFS) formalism is a rigorous approach of modelling multi-object tracking problems and was introduced by Mahler in 2004 [4]. Further, the textbooks [3] and [18] represent the complete mathematical derivations of the RFS concept and its calculus described as finite set statistics (FISST). Mahler comments, that FISST is an "engineering friendly formulation of point process theory" [31, p. 1156]. The key element is the introduction of sets that contain a random number of single object states, each represented by itself as a random vector. In a RFS the number of object states is random and in the set notation the position within the set does not play a role, i.e. $\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$ implies the same as $\{\boldsymbol{x}_2, \boldsymbol{x}_1\}$. Further, the dynamics of each object can vary according a defined motion model and the number of the objects in the set can vary according to a point process model [20]. It can be seen that within this notation the essential of the multi-object tracking problem, estimating the object states and their number, is captured. The tracking problem consists of the computation of the posterior density $p(\boldsymbol{X}_k|\boldsymbol{Z}^k)$, which is the global posterior density of the set-valued quantity.
A RFSs-based filter approaches the estimation problem by using Bayes' theorem to propagate the multi-object density in time, see (3.5), in the same way as a standard single object filter.

$$\cdots \overset{U}{\rightarrow} p(\boldsymbol{X}_{k-1}|\boldsymbol{Z}^{k-1}) \overset{P}{\rightarrow} (\boldsymbol{X}_k|\boldsymbol{Z}^{k-1}) \overset{U}{\rightarrow} p(\boldsymbol{X}_k|\boldsymbol{Z}^k) \overset{P}{\rightarrow} \cdots \tag{3.5}$$

The difference is that the density in the multi-object case is defined on sets and not on vectors or matrices like in the single object case. To handle this, the use of finite set statistics (FISST) becomes necessary. In FISST the underlying idea is to transform a multi-object problem into a mathematically equivalent single object problem as illustrated in Figure 1. Further, the belief-mass, as equivalent to probability mass function in the single object case, is introduced. This is the required function whose derivative gives rise to multi-object set-based Markov densities and likelihood functions.

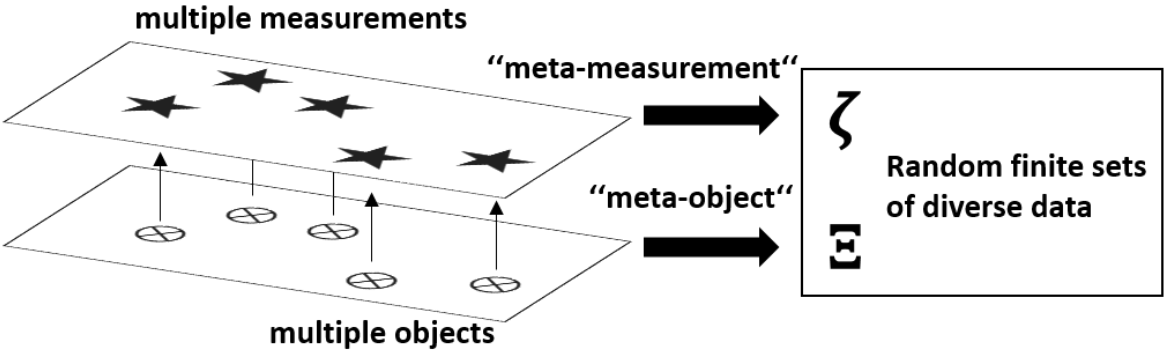

Figure 1: Basic concept of FISST. Multiple measurements of the sensor are transformed in a “meta-measurement”, all objects in a ”meta-object”, both represented by RFSs.

3.3.1. Finite set statistics

A RFS is defined, according to [3], as “a random variable Ξ that draws its instantiations $\Xi = \boldsymbol{X}$ from a hyperspace[2] $\mathcal{X}$ of all finite subsets $\boldsymbol{X}$ (the null set $\emptyset$ included) of some underlying space $\mathcal{X}_0$.” Expressed more intuitively, a RFS is a set whose members are random variables (in the context of tracking random vectors) and whose cardinality is a random variable. The underlying space $\mathcal{X}_0$ could be a state space or in case of a measurement RFS the measurement space with less dimensions. Note that the underlying space $\mathcal{X}_0$ can be any space but in this thesis is restricted to the Euclidean space with $\mathcal{X}_0 = \mathbb{R}^{n_x}$ in case of a RFS that contains object states. The hyperspace $\mathcal{X}$ includes all finite subsets, where every subset $\boldsymbol{X}$ is a instantiations of the RFS Ξ. So the hyperspace $\mathcal{X}$ contains the following subsets:

$$\begin{aligned}
&\boldsymbol{X} = \emptyset \\
&\boldsymbol{X} = \boldsymbol{x}^{(1)} \in \mathbb{R}^{n_x} \\
&\vdots \\
&\boldsymbol{X} = \left\{\boldsymbol{x}^{(1)}, \dots . \boldsymbol{x}^{(n)}\right\}, \qquad \boldsymbol{x}^{(i)} \in \mathbb{R}^{n_x}, i = 1, \dots, n \\
&\vdots
\end{aligned} \tag{3.6}$$

The set cardinality of a finite set $\boldsymbol{X}$ is $|\boldsymbol{X}| = n$, where n is a random variable as already mentioned. A simple interpretation of the hyperspace is that there can be any number of objects and according to (3.6) each object state is a vector in $\mathbb{R}^{n_x}$.

As mentioned in the introduction of section 3.3 the underlying idea of FISST is to define the mathematical equivalents according to the single object case, where it is sufficient to deal with random variables or random vectors. Thus, two important functions, or statistical descriptors, are the probability mass function and the probability density function. The probability mass function $P_{\boldsymbol{x}}(S)$ of a random variable or vector $\boldsymbol{x} \in \mathcal{X}_0$ describes the probability of $\boldsymbol{x}$ being in the subspace $S \subseteq \mathcal{X}_0$,

$$P_{\boldsymbol{x}}(S) = \Pr(\boldsymbol{x} \in S). \tag{3.7}$$

The probability density function $p_{\boldsymbol{x}}(x)$ gives the relative likelihood of $\boldsymbol{x}$ to exist at a given point x. When the probability density function (pdf) is integrated over the subspace $S \subseteq \mathcal{X}_0$, then the probability mass function (pmf) is obtained as given in

[2] In mathematical context, a hyperspace is any space whose points are subsets of another space.

$$P_x(S) = \int_S p_x(\boldsymbol{x})d\boldsymbol{x}. \qquad (3.8)$$

Further, pdf and pmf are related via a derivative defined as

$$p_x(x) = \frac{dP_x(S)}{d\boldsymbol{x}} \mid_{S=x} . \qquad (3.9)$$

Both the pmf and the pdf contain equivalent information and if one of them is available the corresponding one can be derived via differentiation or integration.

In the context of the multi-object case when using RFSs, Mahler describes the generalizations of (3.8) and (3.9) in [3] as belief-mass functions β_Ξ (bmf) and as multi-object probability density functions (mpdf) p_Ξ. Additionally, a third statistical descriptor is mentioned, the probability generating functionals G_Ξ (p.g.fl.), whereas the latter is not further discussed here.

On the one hand the equivalent for the pmf of a random vector or random variable is the bmf for RFSs denoted as $\beta_\Xi(S)$. It defines the probability that the RFS Ξ on the underlying space $\mathcal{X}_0$ is within any measurable S,

$$\beta_\Xi(S) = \Pr(\mathit{\Xi} \subseteq S). \qquad (3.10)$$

On the other hand the complement of the pdf of a random vector or random variable is the mpdf $p_\Xi(\boldsymbol{X})$ of a RFS Ξ.

A general multi-object density function $p(\boldsymbol{X})$ is a real valued function of a finite subset variable $\boldsymbol{X} \subseteq \mathcal{X}_0$. The multi-state density function $p(\boldsymbol{X})$ is a real multi-object probability density function if

$$p(\boldsymbol{X}) \geq 0, \forall \boldsymbol{X} \qquad (3.11)$$

and if

$$\int p(\boldsymbol{X})\delta\boldsymbol{X} = 1. \qquad (3.12)$$

Thus the mpdf $p_\Xi(\boldsymbol{X})$ of an RFS Ξ, if it exists, states

$$\int_S p_\Xi(\boldsymbol{X})\delta\boldsymbol{X} = \Pr(\Xi \subseteq S)\,, \forall S. \qquad (3.13)$$

The relation between the mpdf $p_\Xi(\boldsymbol{X})$ and the bmf $\beta_\Xi(S)$ is thus given by the so called set integral

$$\beta_\Xi(S) = \int_S p_\Xi(\boldsymbol{X})\delta\boldsymbol{X} \qquad (3.14)$$

and its inversion, the so called set derivative [15]

$$p_\Xi(\boldsymbol{X}) = \frac{\delta\beta_\Xi(S)}{\delta\boldsymbol{X}} \mid_{S=\emptyset}. \qquad (3.15)$$

In general, a set integral as in (3.14) is defined as [32, p. 381]

$$\int p(\boldsymbol{X})\delta\boldsymbol{X} \triangleq p(\emptyset) + \sum_{n=1}^{\infty} \frac{1}{n!} \int_{S^n} p(\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}\})d\boldsymbol{x}^{(1)} \ldots d\boldsymbol{x}^{(n)}. \qquad (3.16)$$

For the definition of p.g.fl.s and more information about the pmf and pdf as well as set derivatives, the reader is referred to chapter 11 in [3, p. 343]. The functions in (3.10) and (3.15) have major importance because the pmfs $\beta_\Xi(S)$ are used to construct true multi-object likelihood functions from multi-object measurement models, and multi-object Markov density functions from multi-object motion models. Further, the mpdfs $p_\Xi(\boldsymbol{X})$ contain all information about object numbers and their states and are propagated in time during the filtering process.

Since in set notation the order of elements does not matter, it follows for the simple case of a RFS distribution with two states that

$$p_\Xi\left(\left\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}\right\}\right) = p_\Xi\left(\left\{\boldsymbol{x}^{(2)}, \boldsymbol{x}^{(1)}\right\}\right) = 2p_{\boldsymbol{x}}\left(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}\right) \tag{3.17}$$

which means that the probability density must be distributed equally over the two set possibilities $\left\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}\right\}$ and $\left\{\boldsymbol{x}^{(2)}, \boldsymbol{x}^{(1)}\right\}$. In general for a RFS with n states it applies that

$$p_\Xi\left(\left\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(n)}\right\}\right) = n!\, p_{\boldsymbol{x}}\left(\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(n)}\right), \tag{3.18}$$

where $p_{\boldsymbol{x}}$ is called symmetric joint distribution. As mention below (3.6) the cardinality of a RFS Ξ is a random variable and can be modelled with a specific distribution. The cardinality distribution of a RFS Ξ with n as the number of members is [3, p. 363]

$$\begin{aligned} p_\Xi(n) &\triangleq p_{|\Xi|}(n) \\ &= \Pr(|\Xi| = n) \\ &= \frac{1}{n!}\int_{\mathcal{X}_0} p\left(\left\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(n)}\right\}\right) d\boldsymbol{x}^{(1)} \dots d\boldsymbol{x}^{(n)}. \end{aligned} \tag{3.19}$$

Let $\rho(n)$ be a probability distribution on the non-negative integers and let $p_{\boldsymbol{x}}(\boldsymbol{x})$ be a probability density function on the underlying space $\mathcal{X}_0$. For any $\boldsymbol{X} = \left\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(n)}\right\}$ with the cardinality $|\boldsymbol{X}| = n$, define

$$p_\Xi(\boldsymbol{X}) \triangleq n!\, \rho(n) \prod_{i=1}^{n} p_{\boldsymbol{x}}\left(\boldsymbol{x}^{(i)}\right). \tag{3.20}$$

The RFS in (3.20) is called a i.i.d. cluster RFS, because given the cardinality $|\boldsymbol{X}|$ the elements $\boldsymbol{x}$ of that RFS are each independent identically distributed (i.i.d.) random vectors with a spatial distribution $p_{\boldsymbol{x}}(\boldsymbol{x})$ on $\mathcal{X}_0$.

An important i.i.d. cluster RFS is the Poisson RFS. This applies when $\rho(n)$ is the Poisson pmf, i.e. the number of elements is Poisson distributed as

$$\begin{aligned} \rho(n) &= \mathcal{PS}(n; \gamma) \\ &= \frac{\gamma^n}{n!} e^{-\gamma}, \end{aligned} \tag{3.21}$$

where γ is the mean or rate of the Poisson pmf. The subsequent subset $\boldsymbol{X}$ with pdf $p_\Xi(\boldsymbol{X})$, when inserting (3.21) into (3.20), results as

$$p_\Xi(\boldsymbol{X}) = \gamma^n e^{-\gamma} \prod_{i=1}^{n} p_{\boldsymbol{x}}\left(\boldsymbol{x}^{(i)}\right), \tag{3.22}$$

where n is the number of members in the finite set. (3.32) is a multi-object Poisson process, which is used in the GIW-PHD filter to model the cardinality of the objects and the number of received measurements.

Another important RFS process is the Bernoulli RFS process, where the probability of $\boldsymbol{X}$ being empty is $1 - r$. Vice versa the probability of containing exactly one element $\boldsymbol{x}$ is r. Subsequently the pdf $p_\Xi(\boldsymbol{X})$ for the Bernoulli RFS is

$$p_\Xi(\boldsymbol{X}) = \begin{cases} 1 - r, & \boldsymbol{X} = \emptyset \\ r \cdot p_{\boldsymbol{x}}(\boldsymbol{x}), & \boldsymbol{X} = \{\boldsymbol{x}\}. \end{cases} \tag{3.23}$$

As it can be seen, a single Bernoulli RFS can capture at most one object. For a multi-object scenario, the RFS can be extended to a multi-Bernoulli RFS that is a union of independent Bernoulli's, see e.g. (3.35).

3.3.2. Multi-object Bayes filter

In section 3.3.1 the concept of RFS, its calculus the FISST and some RFS processes were addressed. With these tools a recursive multi-object Bayes filter can be derived using Bayes' theorem as in the single object case, cf. chapter 2.3. Note that for simplicity in the remainder of this chapter the multi-object pdf is simplified to

$$p(\boldsymbol{X}) \triangleq p_{\Xi}(\boldsymbol{X}). \tag{3.24}$$

Assuming that a posterior distribution for the multi-object set is given at time step $k-1$ as

$$p\left(\boldsymbol{X}_{k-1} \middle| \boldsymbol{Z}^{k-1}\right), \tag{3.25}$$

this distribution can be predicted by doing the Bayes prediction step for the multi-object case [31], equivalent to (2.19),

$$p\left(\boldsymbol{X}_k \middle| \boldsymbol{Z}^{k-1}\right) = \int p(\boldsymbol{X}_k | \boldsymbol{X}_{k-1}) p\left(\boldsymbol{X}_{k-1} \middle| \boldsymbol{Z}^{k-1}\right) \delta \boldsymbol{X}_{k-1}. \tag{3.26}$$

The distribution $p(\boldsymbol{X}_k | \boldsymbol{X}_{k-1})$ is the a priori multi-object Markov transition density, which describes the probability of the objects, assuming that they will form the state set $\boldsymbol{X}_k$ at time step k, when they had state set $\boldsymbol{X}_{k-1}$ at time step $k-1$.

The update step, equivalent to (2.20), is

$$p\left(\boldsymbol{X}_k \middle| \boldsymbol{Z}^k\right) = \eta_M \, p(\boldsymbol{Z}_k | \boldsymbol{X}_k) \, p\left(\boldsymbol{X}_k \middle| \boldsymbol{Z}^{k-1}\right) \tag{3.27}$$

where the normalization factor $\eta_M = \frac{1}{p(\boldsymbol{Z}_k | \boldsymbol{Z}^{k-1})}$ has to be calculated using the law of total probability by integrating over all state sets that could have caused the given measurement set:

$$p\left(\boldsymbol{Z}_k \middle| \boldsymbol{Z}^{k-1}\right) = \int p(\boldsymbol{Z}_k | \boldsymbol{X}_k) p\left(\boldsymbol{X}_k \middle| \boldsymbol{Z}^{k-1}\right) \delta \boldsymbol{X}_k. \tag{3.28}$$

In (3.27) $p(\boldsymbol{Z}_k | \boldsymbol{X}_k)$ is the a priori multi-object measurement likelihood. This models the probability that a sensor will receive the measurement set $\boldsymbol{Z}_k$ at time step k if the objects with state set $\boldsymbol{X}_k$ are present.

It can be seen that in (3.25)-(3.28) of the multi-object Bayes filtering an analogy exists to the single object Bayes filtering. The single object Bayes filter is computationally tractable, whereas its generalization, the multi-object Bayes filter, is not. This is because in (3.26) and (3.28) one has to compute set integrals, defined in (3.16). To solve this issue in the next subsections an approximation of the multi-object Bayes filter is introduced.

3.3.3. First order moment approximation

One approximation of the multi-object Bayes filter is the so called probability hypothesis density (PHD) filter that uses, as the name implies, the PHD as approximation for the multi-object density. The name PHD was introduced by Mahler [3] in the context of object tracking. In point process theory this first order moment is known as intensity function, thus the PHD is the statistical first order moment of a multi-object pdf.

For a simple random variable $\boldsymbol{x} \in \mathbb{R}^{n_x}$ with conditional pdf $p_x(\boldsymbol{x}_k | \boldsymbol{z}_k)$ the first order moment is called expectation value and is defined as

$$\begin{aligned}\mathbb{E}_{p_x}\{\boldsymbol{x}_k|\boldsymbol{z}_k\}^3 &\triangleq \int \boldsymbol{x}_k\, p_x(\boldsymbol{x}_k|\boldsymbol{z}_k)d\boldsymbol{x}_k \\ &= \widehat{\boldsymbol{x}}_k.\end{aligned} \tag{3.29}$$

For a multi-object density $p(\boldsymbol{X}_k|\boldsymbol{Z}^k)$ the first order moment is the PHD, denoted $D_{k|k}(\boldsymbol{x}_k|\boldsymbol{Z}^k)$, that is defined on the single object state $\boldsymbol{x} \in \mathcal{X}_0$ given as [32]

$$\begin{aligned}D_{k|k}(\boldsymbol{x}_k|\boldsymbol{Z}^k) &= \int \delta_{\boldsymbol{X}}(\boldsymbol{x}_k) \cdot p(\boldsymbol{X}_k|\boldsymbol{Z}^k)\,\delta\boldsymbol{X}_k \\ &= \int p(\{\boldsymbol{x}_k\} \cup \boldsymbol{Y}_k|\boldsymbol{Z}^k)\delta\boldsymbol{Y}_k,\end{aligned} \tag{3.30}$$

where $\delta_{\boldsymbol{X}}(\boldsymbol{x}_k) \triangleq \sum_{\boldsymbol{y}_k \in \boldsymbol{X}_k} \delta_{\boldsymbol{y}}(\boldsymbol{x}_k)$[4] is a suitable substitution for $\boldsymbol{X}_k$ when reformulating the definition of the expectation value in (3.29) for a multi-object density. It can be stated that the PHD for a RFS is as the expectation value for a random variable. For the whole derivation using the belief-mass function β_Ξ or p.g.fl.s G_Ξ see [3, pp. 576-577, 580-582].
The PHD is completely characterized by the property, given any $S \subseteq \mathcal{X}_0$, that the integral over S results in the expected number of objects $n_{k|k}$ as

$$\int_S D_{k|k}(\boldsymbol{x})d\boldsymbol{x} = \mathbb{E}\{|\boldsymbol{X} \cap S|\} = n_{k|k}(S). \tag{3.31}$$

The total number of objects $N_{k|k}$ when S spans over the complete object state space or scenario, i.e. $S \subset \mathcal{X}_0$, is defined as [3]

$$N_{k|k} \triangleq \int D_{k|k}(\boldsymbol{x})d\boldsymbol{x}. \tag{3.32}$$

3.4. Multi-object state transition and measurement modelling

The FISST calculus allows to model the multi-object states and the measurements using RFS. In the sections above the modelling of the multi-objects state $\boldsymbol{X}_k$ and the sensor collected measurements $\boldsymbol{Z}_k$ was not further addressed. This will be done in this section through considering the assumptions of section 3.1 of the multi-object tracking problem.
The multi-object transition $p(\boldsymbol{X}_k|\,\boldsymbol{X}_{k-1})$ can be described with a Markov multi-object transition density [27] as

$$p(\boldsymbol{X}_k|\,\boldsymbol{X}_{k-1}) = \phi_{k|k-1}(\boldsymbol{X}_k|\,\boldsymbol{X}_{k-1}), \tag{3.33}$$

where the function $\phi_{k|k-1}$, which is described here, incorporates object motion, birth, death and spawn. For a multi-object state $\boldsymbol{X}_{k-1}$ at time step $k-1$ each involved state $\boldsymbol{x}_{k-1} \in \boldsymbol{X}_{k-1}$ continues to exist in the next time step k with a survival probability $p_{S,k}(\boldsymbol{x}_{k-1})$ and then is processed via a single object transition $f(\cdot)$ as defined in (2.5) or dies with the probability $1 - p_{S,k}(\boldsymbol{x}_{k-1})$. This survival or death process is modelled as Bernoulli RFS [27] denoted as

$$\boldsymbol{S}_{k|k-1}(\{\boldsymbol{x}_{k-1}\}) = \begin{cases} 1 - p_{S,k}(\boldsymbol{x}_{k-1}), & \boldsymbol{X}_{k-1} = \emptyset \\ p_{S,k}(\boldsymbol{x}_{k-1})f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}), & \boldsymbol{X}_{k-1} = \{\boldsymbol{x}_{k-1}\}. \end{cases} \tag{3.34}$$

Further, at time step k a new object can appear due to either spawning from an existing object $\boldsymbol{B}_{k|k-1}(\cdot)$ or due to a spontaneous birth $\boldsymbol{\Gamma}_k$ that is independent of existing objects. Both cases are modelled as RFSs. The consisting RFS that incorporates the independent union of the described actions above states as

[3] $\mathbb{E}\{\cdot\}$ is the expectation operator.

[4] $\delta_{\boldsymbol{y}}(\boldsymbol{x})$ is the Dirac delta function concentrated at $\boldsymbol{y}$.

$$\boldsymbol{X}_k = \left[\bigcup_{x\in\boldsymbol{X}_{k-1}} \boldsymbol{S}_{k|k-1}(\boldsymbol{x})\right] \cup \left[\bigcup_{x\in\boldsymbol{X}_{k-1}} \boldsymbol{B}_{k|k-1}(\boldsymbol{x})\right] \cup \boldsymbol{\Gamma}_k. \tag{3.35}$$

Note that modelling of the spontaneous birth process $\boldsymbol{\Gamma}_k$ and the spawning $\boldsymbol{B}_{k|k-1}(\cdot)$ is problem dependent. An example for the birth process is given in chapter 8. For spawning see e.g. [12].
The multi-object measurement likelihood $p(\boldsymbol{Z}_k|\,\boldsymbol{X}_k)$ can be described by the multi-object observation model

$$p(\boldsymbol{Z}_k|\,\boldsymbol{X}_k) = \varphi_{k|k-1}(\boldsymbol{Z}_k|\,\boldsymbol{X}_k), \tag{3.36}$$

which incorporates the detection of an object with a certain probability and the reception of clutter measurements. Each state $\boldsymbol{x}_k \in \boldsymbol{X}_k$ at time step k is detected with a probability of detection $p_{D,k}(\boldsymbol{x}_k)$ or missed with $1 - p_{D,k}(\boldsymbol{x}_k)$ and therefore either creates a measurement RFS $\boldsymbol{\Theta}_k(\boldsymbol{x}_k)$ or not. Equivalent to (3.34) $\boldsymbol{\Theta}_k(\cdot)$ is a Bernoulli RFS noted as

$$\boldsymbol{\Theta}_k(\{\boldsymbol{x}_k\}) = \begin{cases} 1 - p_{D,k}(\boldsymbol{x}_k), & \boldsymbol{Z}_k = \emptyset \\ p_{D,k}(\boldsymbol{x}_k)h(\boldsymbol{z}_k|\boldsymbol{x}_k), & \boldsymbol{Z}_k = \{\boldsymbol{z}_k\}, \end{cases} \tag{3.37}$$

where the measurement process is captured by $h(\cdot)$ as defined in (2.7). Further, the false alarms or clutter measurements, which the sensor receives at time step k, are noted as set $\boldsymbol{K}_k$ so that the resulting measurement RFS $\boldsymbol{Z}_k$ is an independent union of measurements and clutter given as

$$\boldsymbol{Z}_k = \left[\bigcup_{x\in\boldsymbol{X}_k} \boldsymbol{\Theta}_k(\boldsymbol{x})\right] \cup \boldsymbol{K}_k. \tag{3.38}$$

The modelling of clutter measurements $\boldsymbol{K}_k$ is problem dependent. A common implementation is to model the number of clutter measurements as Poisson distributed and its spatial spreading over the surveillance area with a uniform distribution, see chapter 8.
An multiple object tracking algorithm is given by Vo et al. [12], called the Gaussian mixture PHD filter. This filter is presented in section 3.5 and uses for the motion model in (3.34) and measurement model in (3.37) the respective point object variants given in (2.5) and (2.7). Other explicit expressions for these models can be derived using FISST [3] [18], but are not necessary here.

3.5. The Gaussian mixture PHD filter

The PHD filter as an approximation of the multi-object Bayes filter propagates only the PHD (3.30) as first order moment in time:

$$\cdots \overset{U}{\rightarrow} D_{k-1|k-1}(\boldsymbol{x}) \overset{P}{\rightarrow} D_{k|k-1}(\boldsymbol{x}) \overset{U}{\rightarrow} D_{k|k}(\boldsymbol{x}) \overset{P}{\rightarrow} \cdots \tag{3.39}$$

For this time propagation an initial PHD $D_{k-1|k-1}(\boldsymbol{x})$, a PHD predictor $D_{k|k-1}(\boldsymbol{x})$ and a PHD corrector $D_{k|k}(\boldsymbol{x})$ are needed. The derivation of those was done by Mahler and can be reviewed in [3] or [31], where the models of object transition and measurement introduced in section 3.4 are taken into account. For the presentation of the filter equations some essential assumptions have to be considered:

A.1. The objects are independent of one another in respect of measurement generation and motion evolution.

A.2. The clutter RFS is modelled as a Poisson process and clutter measurements are independent of object originated measurements.

A.3. The predicted multi-object RFS $p(\boldsymbol{X}_k|\boldsymbol{Z}^{k-1})$ is a Poisson RFS.

Note that assumptions A.1 and A.2 are common in many tracking scenarios [1], whereas A.3 can be seen as a reasonable approximation if the interaction of objects is negligible as remarked in [31].

Let $D_{k|k-1}(\boldsymbol{x})$ and $D_{k|k}(\boldsymbol{x})$ denote the respective PHDs associated with the predicted multi-object density $p(\boldsymbol{X}_k|\boldsymbol{Z}^{k-1})$ and the posterior multi-object density $p(\boldsymbol{X}_k|\boldsymbol{Z}^k)$ in (3.26) and (3.27). Under the assumption made above the PHD recursion can be derived using FISST [31] or classical probabilistic calculus as remarked in [12], thus is given as

$$\begin{aligned} D_{k|k-1}(\boldsymbol{x}_k) = \gamma(\boldsymbol{x}_k) + \int p_{S,k}(\boldsymbol{x}_{k-1}) f_{k|k-1}(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})\, D_{k-1}(\boldsymbol{x}_{k-1})\delta\boldsymbol{x}_{k-1} \\ + \int \beta_{k|k-1}(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) D_{k-1}(\boldsymbol{x}_{k-1}) d\boldsymbol{x}_{k-1} \end{aligned} \tag{3.40}$$

and

$$\begin{aligned} D_{k|k}(\boldsymbol{x}_k) = \left(1 - p_{D,k}(\boldsymbol{x}_k)\right) D_{k|k-1}(\boldsymbol{x}_k) \\ + \sum_{\boldsymbol{z}\in\boldsymbol{Z}_k} \frac{p_{D,k}(\boldsymbol{x}_k) h_k(\boldsymbol{z}|\boldsymbol{x}_k) D_{k|k-1}(\boldsymbol{x}_k)}{\kappa_k(\boldsymbol{z}) + \int p_{D,k}(\boldsymbol{x}') + h_k(\boldsymbol{z}|\boldsymbol{x}') D_{k|k-1}(\boldsymbol{x}') dx'}, \end{aligned} \tag{3.41}$$

where

- $\gamma_k(\cdot)$ is the birth intensity of the RFS $\boldsymbol{\Gamma}_k$ (3.35) at time step k
- $\beta_{k|k-1}(\cdot\mid\cdot)$ is the spawning intensity of the RFS $\boldsymbol{B}_{k|k-1}(\cdot)$ (3.35) at time step k
- $p_{S,k}(\cdot)$ and $p_{D,k}(\cdot)$ are the survival probability and detection probability at time step k, defined in section 3.4
- $\kappa_k(\cdot)$ is the clutter intensity of the clutter RFS $\boldsymbol{K}_k$ (3.38) at time step k.

It can be seen that the PHD recursion, (3.40) and (3.41), completely avoids an explicit measurement to object association, which is typical for a RFS filter approach. Further, the posterior PHD $D_{k|k}(\cdot)$ is a function defined on the single object space $\mathcal{X}_0 = \mathbb{R}^{n_x}$, whereas the multi-object Bayes recursion (3.26) and (3.27) operates on the hyperspace $\mathcal{X}$, thus the PHD recursion is computational less expensive.

To implement the PHD filter in real world applications a closed form solution is provided by Vo et al. in 2006 [12] that uses Gaussian mixtures (GMs). The presented filter is derived by the authors under three additional assumptions added to A.1 - A.3:

A.4. Each object's dynamic can be described using a linear Gaussian motion model, equivalent to (2.27), and the measurement process follows a linear Gaussian measurement model, equivalent to (2.28):

$$f_{k|k-1}(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) = \mathcal{N}(\boldsymbol{x}_k; F\boldsymbol{x}_{k-1}, Q_{k-1}), \tag{3.42}$$

$$h_k(\boldsymbol{z}_k|\boldsymbol{x}_k) = \mathcal{N}(\boldsymbol{z}_k; H\boldsymbol{x}_k, R_k). \tag{3.43}$$

A.5. The survival and detection probabilities are state independent, thus

$$p_{S,k}(\boldsymbol{x}) = p_{S,k}, \tag{3.44}$$

$$p_{D,k}(\boldsymbol{x}) = p_{D,k}. \tag{3.45}$$

A.6. The spawn and birth RFSs are intensities modelled with Gaussian mixtures in form of

$$\gamma_k(\boldsymbol{x}_k) = \sum_{i=1}^{J_{\gamma,k}} w_{\gamma,k}^{(i)} \mathcal{N}\left(\boldsymbol{x}_k; \boldsymbol{m}_{\gamma,k}^{(i)}, P_{\gamma,k}^{(i)}\right) \tag{3.46}$$

and

$$\beta_{k|k-1}(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) = \sum_{j=1}^{J_{\beta,k}} w_{\beta,k}^{(i)} \mathcal{N}\left(\boldsymbol{x}_k; F_{\beta,k-1}^{(j)} \boldsymbol{x}_{k-1} + \boldsymbol{d}_{\beta,k-1}^{(j)}, Q_{\beta,k-1}^{(j)}\right), \tag{3.47}$$

where $J_{\gamma,k}$, $w_{\gamma,k}^{(i)}$, $\boldsymbol{m}_{\gamma,k}^{(i)}$, $P_{\gamma,k}^{(i)}$ with $i \ldots J_{\gamma,k}$ are parameter for modelling the shape of the birth intensity. Similarly $J_{\beta,k}$, $w_{\beta,k}^{(j)}$, $F_{\beta,k-1}^{(j)}$, $\boldsymbol{d}_{\beta,k-1}^{(j)}$, $Q_{\beta,k-1}^{(j)}$ with $j \ldots J_{\beta,k}$ determine the object spawning intensity. The modelling of these two intensities is problem dependent.

With all assumptions A.1 - 0 made for the GM PHD filter, a closed form PHD recursion is derived in [12, p. 5], where the posterior PHD $D_{k-1|k-1}$ is given as GM and the predicted PHD $D_{k|k-1}$ as well. The posterior PHD at time step $k-1$ is denoted as

$$D_{k-1|k-1}(\boldsymbol{x}_{k-1}) = \sum_{i=1}^{J_{k-1}} w_{k-1}^{(i)} \mathcal{N}\left(\boldsymbol{x}_{k-1}; \boldsymbol{m}_{k-1}^{(i)}, P_{k-1}^{(i)}\right). \tag{3.48}$$

Thus, the predicted PHD also is a sum of GMs, consisting of the PHD of survived object $D_{S,k|k-1}$, of spawned objects $D_{\beta,k|k-1}$ and born objects D_γ given as

$$D_{k|k-1}(\boldsymbol{x}_k) = D_{S,k|k-1}(\boldsymbol{x}_k) + D_{\beta,k|k-1}(\boldsymbol{x}_k) + D_\gamma(\boldsymbol{x}_k). \tag{3.49}$$

The calculation of the single components of the GM PHD filter algorithm is given in the Appendix in Table 24. Further, a pruning and merging scheme is provided in the Appendix A.1 in Table 25, as well as the pseudo code for state extraction in Table 26. Pruning and merging has to be taken into account, because otherwise the number of filter components J_k would increase to a level after a few time steps that would not be computationally traceable.

For the sake of completeness, it should be noted here that the presented analytic implementation of the PHD filter is not the only possibility. Discrete implementations using Monte Carlo techniques are also common, see [33], but are computationally much more expensive. Further, one drawback of the PHD filter is the characteristic that the cardinality is not very robust. This is because the object RFS is modelled as Poisson process. An improvement of the classical PHD filter is the cardinalized PHD (CPHD) filter that propagates the cardinality distribution in addition to the PHD in time, see [11]. Further, neither the PHD nor the CPHD filter provide by its basic forms track labels. A track is a trajectory over time that is defined as a sequence of object states of one particular object. Track labels for these filters can be obtained using post-processing schemes as in [34].

4. Extended object tracking

In this chapter, the extended object tracking problem is addressed. The objects in chapter 2 and 3 were assumed to be point objects, generating at most one measurement per time step. Now, the objects are assumed to be so called *extended* objects. In [15, p. 71] it is defined that an extended object "gives rise to more than one measurement per time step". This is the case, when the setup between object and sensor is in a such way that the object occupies more than one resolution cell, e.g. when using laser sensors or high resolution radar sensors in near-field applications. When receiving multiple measurements per object, it becomes possible to create an estimate of both, the kinematic state and the extent state. The latter could include object shape, size and orientation. With these properties it is also possible to distinguish between different types of objects, e.g. car, bicycle or pedestrian. A comprehensive overview of the field of extended object tracking with its large varieties is given in [35].

Section 4.1 starts with the introduction of the assumptions for the extended object tracking problem. In section 4.2, the spatial measurement modelling [13] is introduced, which implies the measurements to be spread according to a spatial distribution. Further, the number of measurements can be modelled according to a cardinality distribution, where in this section the example of a Poisson point process model is given. In section 4.3, the random matrix approach of Koch [7] is discussed with the underlying models, derivations and calculations of the prediction and update step when using the approach in a Bayesian framework.

4.1. The extended object tracking problem

For the extended object tracking problem the assumptions A.1 - A.5, made in section 3.1 associated with multi-object tracking, apply as well. Additionally, one further assumption is made that claims the unknown number of measurements per object. For the sake of completeness all assumptions for the extended object tracking problems are depicted:

A.1. Each measurement that is caused by an object is noise corrupted.

A.2. The detection probability of each object is less than one and therefore it is not known whether an object created a measurement at the current time step or not.

A.3. Beside object originated measurements there exist false alarms or clutter measurements.

A.4. The origin of each measurement is unknown, i.e. one does not know whether one measurement is clutter or object generated.

A.5. The number of existing objects is unknown.

A.6. The number of measurements generated per object is unknown.

Remark: The assumptions A.1 - A.6 are assumed for extended object tracking with multiple objects. For single extended object tracking, as described in this chapter, A.3 and A.5 are dropped. That means, there is only one object and the measurements obtained are without clutter.

Let ξ_k be the extended state vector at time step k, which contains information about both object kinematics and object extension. Using probabilistic tools the extended object tracking problem consists of estimating the extended object state ξ_k out of a set of measurements $\boldsymbol{Z}_k$ at time step k, cf. (3.2). This is done by modelling the conditional distribution

$$p(\xi_k|\boldsymbol{Z}_k) \tag{4.1}$$

and extract a state estimate $\hat{\xi}_k$ at time step k with an appropriate estimator.

4.2. Measurement modelling

In [35] the modelling of the extension is divided in principle into three approaches. First, the set of points on a rigid body (SPRB) model assumes that the object surface is a rigid body with a fix number of reflection points, which can generate measurements. The reflection points are detected independently with a certain probability of detection. The crucial part of this approach is to associate the measurements to the reflection points. This can be done with the common methods presented in section 3.2. A second approach is to use physics based modelling that incorporates the physical modelling of the sensors properties. This approach is highly dependent of the used sensor and the object that has to be detected. A third and more general approach is discussed in this section, called the spatial model presented by Gilholm et al. in [13] and [36]. This model uses an inhomogeneous Poisson point process (PPP) to model the detections that originate from an extended object. That means, an object is represented by a spatial distribution and the object measurements are more likely to stem from regions with high spatial density rather than from low one [13]. The authors further assume in their derivations that the clutter distribution and the spatial extent model are known. In the following paragraphs the measurement model is presented briefly.
To solve the extended object tracking problem defined in section 4.1 in a Bayesian framework, it is necessary to model the a priori distribution of the measurements given a certain object state as

$$p(\boldsymbol{Z}_k|\xi_k). \tag{4.2}$$

The measurement model in (4.2) can be rewritten in dependence of the number of measurements $n_{\mathbf{z}_k}$ that an object generates, denoted as

$$p(\boldsymbol{Z}_k, n_{\mathbf{z}_k}|\xi_k) = p(\boldsymbol{Z}_k|n_{\mathbf{z}_k}, \xi_k)p(n_{\mathbf{z}_k}|\xi_k). \tag{4.3}$$

Thus $p(\boldsymbol{Z}_k|n_{\boldsymbol{Z}_k}, \xi_k)$ is the spatial distribution and $p(n_{\boldsymbol{Z}_k}|\xi_k)$ the cardinality distribution of received measurements. It is common to model the measurements in the spatial distribution as i.i.d. [13], which means

$$p(\boldsymbol{Z}_k|n_{\mathbf{z}_k}, \xi_k) = \prod_{j=1}^{n_{Z_k}} p(\mathbf{z}_k^{(j)}|\xi_k), \tag{4.4}$$

where $p(\mathbf{z}_k^{(j)}|\xi_k)$ is the distribution for an individual object generated measurement. The cardinality distribution of the measurements $p(n_{\boldsymbol{Z}_k}|\xi_k)$ is modelled in [13] as Poisson distributed as

$$\begin{aligned} p(n_{\boldsymbol{Z}_k}|\xi_k) &= \mathcal{PS}(n_{\boldsymbol{Z}_k}; \lambda_k) \\ &= e^{-\lambda_k}\frac{\lambda_k^{n_{Z_k}}}{n_{\boldsymbol{Z}_k}!}, \end{aligned} \tag{4.5}$$

where λ_k is the rate parameter of the Poisson mass function. This approach of [13] is limited to a single extended object, but was further addressed for multiple objects in [36]. In this thesis, the spatial distribution of the measurements is given as a Gaussian distribution in the GIW-PHD filter. This approach was proposed by Koch and is introduced in the next section.

4.3. Tracking with the random matrix approach

Modelling the shape can be done with many different geometrical approaches. For further details the reader is referred to [35]. A very common approach is to model an object shaped as ellipsis. Associated with that are two common approaches, known as random matrix (RM)

framework and random hypersurface model (RHM). These two are compared by Baum et al. in [37], whereas in this thesis the RM approach will be discussed, because it is used in the filter implementation.
The RM framework was originally proposed by Koch in [7], where the extended object state ξ_k is a combination of a kinematic state vector $\boldsymbol{x}_k$ and extent matrix[5] X_k. The extended object state is defined as

$$\xi_k = (\boldsymbol{x}_k, X_k). \tag{4.6}$$

Koch considers the object extension as an “additional internal degree of freedom” [7, p. 1042] that jointly has to be estimated with the kinematics state. This is done using the Bayesian filtering framework: The extension at time step k is mathematically described by a symmetric positive definite (SPD) matrix X_k that has to be estimated through the sensor measurements. In the original work [7] extended objects are assumed to be well separated from each other, so that inter-cluster data associations are avoided. The joint density $p(\xi_k|\boldsymbol{Z}^k) = p(\boldsymbol{x}_k, X_k|\boldsymbol{Z}^k)$ that has to be evaluated is a product of Gaussian- and Wishart-related densities [38] stated as

$$p(\boldsymbol{x}_k, X_k|\boldsymbol{Z}^k) = p(\boldsymbol{x}_k|X_k, \boldsymbol{Z}^k)p(X_k|\boldsymbol{Z}^k), \tag{4.7}$$

where $p(\boldsymbol{x}_k|X_k, \boldsymbol{Z}^k)$ is the vector variate pdf and $p(X_k|\boldsymbol{Z}^k)$ is the matrix variate pdf. For an extended object tracking filter the calculation of (4.7) is done in two steps, the prediction (P) and the update (U), visualized as

$$\ldots \overset{U}{\rightarrow} p(\boldsymbol{x}_{k-1}, X_{k-1}|\boldsymbol{Z}^{k-1}) \overset{P}{\rightarrow} p(\boldsymbol{x}_k, X_k|\boldsymbol{Z}^{k-1}) \overset{U}{\rightarrow} p(\boldsymbol{x}_k, X_k|\boldsymbol{Z}^k) \overset{P}{\rightarrow} \ldots \tag{4.8}$$

4.3.1. Motion and measurement model

The originally proposed motion and measurement models by Koch [7] involve linear models with a notation using the so called Kronecker product operator $\otimes$ [6].
The kinematic state vector $\boldsymbol{x}_k$ at time step k is given by

$$\boldsymbol{x}_k = \begin{pmatrix} \boldsymbol{r}_k \\ \dot{\boldsymbol{r}}_k \\ \ddot{\boldsymbol{r}}_k \end{pmatrix}, \tag{4.9}$$

where $\boldsymbol{r}_k$ is the spatial state component, $\dot{\boldsymbol{r}}_k$ the corresponding velocity and $\ddot{\boldsymbol{r}}_k$ the corresponding acceleration. The dimension of vector $\boldsymbol{r}_k$ is denoted as d, thus the extent matrix X_k is a $d \times d$ SPD matrix that describes the current object extent as ellipsis. The dimension of the state vector $\boldsymbol{x_k}$ is of dimension $s \cdot d$, where $s - 1$ describes the order of derivation up to which the object's kinematic is modelled. In Koch's model $s = 3$, i.e. $\boldsymbol{x}_k$ contains position, velocity and acceleration. Thus the state vector in 2-dimensional space with $d = 2$ is defined as

$$\boldsymbol{x}_k = \begin{pmatrix} x \\ y \\ v_x \\ v_y \\ a_x \\ a_y \end{pmatrix}_k, \tag{4.10}$$

where x and y define the Cartesian position, with velocity $v_{(\cdot)}$ and acceleration $a_{(\cdot)}$ in each dimension.

[5] See Gupta and Nagar [38] as reference for matrix variate distributions.

[6] For definition see appendix A.2

The object motion is modelled equivalent to (2.6) linear with additive noise, but with the use of the Kronecker product and denotes

$$\boldsymbol{x}_k^{(i)} = \left(F_{k|k-1} \otimes I_d\right)\boldsymbol{x}_{k-1}^{(i)} + \boldsymbol{w}_k^{(i)}, \tag{4.11}$$

with the identity matrix $I_d \in \mathbb{R}^{d \times d}$ and i as index for the ith object. Note that the index is needed for the presentation of the GIW-PHD filter in chapter 5. $F_{k|k-1} \in \mathbb{R}^{s \times s}$ is the transition matrix given as

$$F_{k|k-1} = \begin{pmatrix} 1 & T_s(k) & \frac{1}{2}T_s^2(k) \\ 0 & 1 & T_s(k) \\ 0 & 0 & e^{-\frac{T_s(k)}{\theta}} \end{pmatrix}, \tag{4.12}$$

with sampling time $T_s(k)$ defined in (2.9). Completing (4.11), $\boldsymbol{w}_k^{(i)} \in \mathbb{R}^{sd}$ is the process noise vector drawn as zero mean Gaussian distribution with covariance matrix $\Delta_{k|k-1}^{(i)}\mathbb{R}^{3d \times 3d}$ denoted

$$\Delta_{k|k-1}^{(i)} = Q_{k|k-1} \otimes X_k^{(i)}, \tag{4.13}$$

where $Q_{k|k-1}$ is a noise matrix of van Keuk's model [39] that only effects the acceleration part:

$$Q_{k|k-1} = \Sigma^2 \left(1 - e^{-2\frac{T_s(k)}{\theta}}\right) \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{4.14}$$

In (4.14) Σ is the scalar standard deviation of the object's acceleration and θ is the manoeuvre correlation time. A large θ models the object motion in a rather straight way or vice versa a small θ assumes a very agile object motion.

Koch comments the fact that the covariance matrix $\Delta_k^{(i)}$ is dependent on the current object extent X_k with some physical motivated arguments that can be reviewed in [7].

The measurement set at time step k is $\boldsymbol{Z}_k$ with the number $n_{\boldsymbol{z}_k}$ of received measurement, cf. (3.2). The measurement model of a single measurement is defined as

$$\boldsymbol{z}_k^{(j)} = (H_k \otimes I_d)\boldsymbol{x}_k^{(i)} + \boldsymbol{e}_k^{(j)}, \tag{4.15}$$

where $\boldsymbol{e}_k^{(j)} \in \mathbb{R}^d$ is a white Gaussian noise vector with covariance matrix given by the objects extent matrix $X_k^{(i)}$ and the measurement matrix H_k is given as

$$H_k = (1\ 0\ 0). \tag{4.16}$$

The fact that the noise $\boldsymbol{e}_k^{(j)}$ in (4.15) is only depending on the extent matrix $X_k^{(i)}$ implies a non-existence of an additional sensor noise. This is naturally not the case in real world applications. Koch reasons this implication with the assumption that the sensor error is negligible small compared to the object's extent.

The dimension of the measurement vector is determined by the model with the Kronecker notation $H_k \otimes I_d$ in (4.15). The notation picks out the Cartesian position and thus it follows that measurement vector is of dimension $\boldsymbol{z} \in \mathbb{R}^d$.

4.3.2. Random matrix Bayes filter

The predicted joint pdf $p(\boldsymbol{x}_k, X_k|\boldsymbol{Z}^{k-1})$, see (4.8), can be interpreted as marginal density that is calculated by the integration [7]

$$\begin{aligned} &p(\boldsymbol{x}_k, X_k|\boldsymbol{Z}^{k-1}) \\ &= \int\int p(\boldsymbol{x}_k, X_k|\boldsymbol{x}_{k-1}, X_{k-1}, \boldsymbol{Z}^{k-1})\, p(\boldsymbol{x}_{k-1}, X_{k-1}|\boldsymbol{Z}^{k-1}) d\boldsymbol{x}_{k-1} dX_{k-1}. \end{aligned} \tag{4.17}$$

For the derivation of the prediction step a few assumptions are made [7, p. 1044] leading to a Chapman-Kolmogorov prediction step as a product of two independent integrals that can be finally split of. These assumptions are stated and applied in the following paragraphs.
The first term in the integral (4.17) is rewritten with the law of conditional probability as the product

$$\begin{aligned} &p(\boldsymbol{x}_k, X_k|\boldsymbol{x}_{k-1}, X_{k-1}, \boldsymbol{Z}^{k-1}) \\ &\qquad = p(\boldsymbol{x}_k|X_{k,}\boldsymbol{x}_{k-1}, X_{k-1}, \boldsymbol{Z}^{k-1}) p(X_k|\boldsymbol{x}_{k-1}, X_{k-1}, \boldsymbol{Z}^{k-1}). \end{aligned} \tag{4.18}$$

Second, for the kinematics term in (4.18) the Markov assumption is applied so that the simplification

$$p(\boldsymbol{x}_k|X_k\boldsymbol{x}_{k-1}, X_{k-1}, \boldsymbol{Z}^{k-1}) = p(\boldsymbol{x}_k|X_{k,}\boldsymbol{x}_{k-1}) \tag{4.19}$$

holds. Third, for the extension term in (4.18) it is assumed that the object's kinematics state $\boldsymbol{x}_{k-1}$ and previous measurements $\boldsymbol{Z}^{k-1}$ have no impact on the change of the extent, i.e. they do not influence the evolution from X_{k-1} to X_k. This assumption leads to

$$p(X_k|\boldsymbol{x}_{k-1}, X_{k-1}, \boldsymbol{Z}^{k-1}) = p(X_k|X_{k-1}). \tag{4.20}$$

Fourth, the previous update is rewritten as

$$p(\boldsymbol{x}_{k-1}, X_{k-1}|\boldsymbol{Z}^{k-1}) = p(\boldsymbol{x}_{k-1}|X_{k-1}, \boldsymbol{Z}^{k-1}) \cdot p(X_{k-1}|\boldsymbol{Z}^{k-1}) \tag{4.21}$$

and it is assumed that the change of the object extent does not influence the kinematical object state in (4.21), given as

$$p(\boldsymbol{x}_{k-1}|X_{k-1}, \boldsymbol{Z}^{k-1}) = p(\boldsymbol{x}_{k-1}|X_k, \boldsymbol{Z}^{k-1}). \tag{4.22}$$

Inserting (4.22) into (4.21) leads to

$$p(\boldsymbol{x}_{k-1}, X_{k-1}|\boldsymbol{Z}^{k-1}) = p(\boldsymbol{x}_{k-1}|X_k, \boldsymbol{Z}^{k-1}) \cdot p(X_{k-1}|\boldsymbol{Z}^{k-1}). \tag{4.23}$$

Finally, the prediction in (4.17) can be rewritten using (4.18), with the inserted simplifications (4.19) and (4.20), and the previous update in (4.21) resulting in

$$\begin{aligned} p(\boldsymbol{x}_k, X_k|\boldsymbol{Z}^{k-1}) = \int\int &\underbrace{p(\boldsymbol{x}_k|X_k, \boldsymbol{x}_{k-1}) p(X_k|X_{k-1})}_{evolution\ model} \\ &\cdot \underbrace{p(\boldsymbol{x}_{k-1}|X_k, \boldsymbol{Z}^{k-1}) p(X_{k-1}|\boldsymbol{Z}^{k-1})}_{previous\ update} d\boldsymbol{x}_{k-1} dX_{k-1} \end{aligned} \tag{4.24}$$

The left-hand side of (4.24) can be rearranged with the law of conditional probability as

$$p(\boldsymbol{x}_k, X_k|\boldsymbol{Z}^{k-1}) = p(\boldsymbol{x}_k|X_k, \boldsymbol{Z}^{k-1}) p(X_k|\boldsymbol{Z}^{k-1}), \tag{4.25}$$

thus the prediction is a product of two densities, the first one depending on the kinematics state $\boldsymbol{x}$ and the second one depending on the extension state X. By sorting the terms in (4.24) according to this, the whole prediction integral can be split up into two independent prediction integrals:

1. Kinematics prediction with

$$p(\boldsymbol{x}_k|X_k,\boldsymbol{Z}^{k-1}) = \int \underbrace{p(\boldsymbol{x}_k|X_k,\boldsymbol{x}_{k-1})}_{kinematic\ evolution}\ \underbrace{p(\boldsymbol{x}_{k-1}|X_k,\boldsymbol{Z}^{k-1})}_{previous\ update}\,d\boldsymbol{x}_{k-1}. \tag{4.26}$$

2. Extent prediction with

$$p(X_k|\boldsymbol{Z}^{k-1}) = \int \underbrace{p(X_k|X_{k-1})}_{extent\ evolution}\ \underbrace{p(X_{k-1}|\boldsymbol{Z}^{k-1})dX_{k-1}}_{previous\ update}. \tag{4.27}$$

After the prediction step, current sensor data $\boldsymbol{Z}^k$ is processed in the update or filtering step, equivalent to (2.20). Therefore the sensor specific likelihood function $p(\boldsymbol{Z}_k,n_{\boldsymbol{z}_k}|\xi_k) = p(\boldsymbol{Z}_k,n_{\boldsymbol{z}_k}|\boldsymbol{x}_k,X_k)$, according to (4.3), is exploited by Bayes' theorem. The update step thus follows as

$$p(\boldsymbol{x}_k,X_k|\boldsymbol{Z}^k) = \frac{p(\boldsymbol{Z}_k,n_{\boldsymbol{z}_k}|\boldsymbol{x}_k,X_k)p(\boldsymbol{x}_k,X_k|\boldsymbol{Z}^{k-1})}{\int\int p(\boldsymbol{Z}_k,n_{\boldsymbol{z}_k}|\boldsymbol{x}_k,X_k)p(\boldsymbol{x}_k,X_k|\boldsymbol{Z}^{k-1})d\boldsymbol{x}_k dX_k}. \tag{4.28}$$

4.3.3. The random matrix prediction step

After the derivation of the Bayes prediction and Bayes update step it is necessary to approximate the distribution in (4.26) - (4.28). This is done in Koch's approach using Gaussian, Wishart and inverse Wishart distributions. See Appendix A.3 for the distributions' definitions.
In section 4.3.2 it was shown that the prediction of kinematics and extension part can be handled separately. For the kinematics part it is assumed that the density after filtering $p(\boldsymbol{x}_{k-1}|X_k,\boldsymbol{Z}^{k-1})$ is a Gaussian distribution and that this Gaussian is preserved after the prediction step in $p(\boldsymbol{x}_k|X_k,\boldsymbol{Z}^{k-1})$. That means,

$$p(\boldsymbol{x}_{k-1}|X_k,\boldsymbol{Z}^{k-1}) = \mathcal{N}(\boldsymbol{x}_{k-1};\boldsymbol{x}_{k-1|k-1},P_{k-1|k-1}\otimes X_k) \tag{4.29}$$

is the density after the previous filtering step and the predicted density (4.26) is modelled as

$$\begin{aligned} p(\boldsymbol{x}_k|X_k,\boldsymbol{Z}^{k-1}) = \int & \mathcal{N}(\boldsymbol{x}_k;(F_{k|k-1}\otimes I_d)\boldsymbol{x}_{k-1},\Delta_{k|k-1}) \\ & \cdot\mathcal{N}(\boldsymbol{x}_{k-1};\boldsymbol{x}_{k-1|k-1},P_{k-1|k-1}\otimes X_k)d\boldsymbol{x}_{k-1} \end{aligned} \tag{4.30}$$

$$= \mathcal{N}(\boldsymbol{x}_k;\boldsymbol{x}_{k|k-1},P_{k|k-1}\otimes X_k) \tag{4.31}$$

The calculations of the predicted state $\boldsymbol{x}_{k|k-1}$ and covariance $P_{k|k-1}\otimes X_k$ are preformed according to the linear Kalman equations cf. Table 1. The equations for the case when using the Kronecker product notation (4.31) are given in Table 2.
The extension state the density $p(X_{k-1}|\boldsymbol{Z}^{k-1})$ in (4.27) is given as an inverse Wishart distribution stated as

$$p(X_{k-1}|\boldsymbol{Z}^{k-1}) = \mathcal{IW}(X_{k-1};\nu_{k-1|k-1},X_{k-1|k-1}), \tag{4.32}$$

where $\nu_{k-1|k-1}$ is the degree of freedom and $X_{k-1|k-1}$ is the inverse scale matrix of the density. Further, the extent evolution density $p(X_k|X_{k-1})$ in (4.27) is assumed as Wishart density as

$$p(X_k|X_{k-1}) = \mathcal{W}\left(X_k;\delta_{k|k-1},\frac{X_{k-1}}{\delta_{k|k-1}}\right), \tag{4.33}$$

where $\delta_{k|k-1} > 0$ is a noise parameter that governs the noise of the prediction. The smaller $\delta_{k|k-1}$, the higher the process noise, i.e. a large change of the ellipse is assumed. Now the extent prediction density in (4.27) can be calculated by integration over the product of (4.32) and (4.33), leading to

$$p(X_k|\boldsymbol{Z}^{k-1}) = \int \mathcal{W}\left(X_k; \delta_{k|k-1}, \frac{X_{k-1}}{\delta_{k|k-1}}\right) \cdot \mathcal{IW}(X_{k-1}; \nu_{k-1|k-1}, V_{k-1|k-1}) dX_{k-1}. \tag{4.34}$$

The resulting density in (4.34) would be a generalized Beta type II density [7, p. 1047], however can be approximated with an inverse Wishart distribution as given

$$p(X_k|\boldsymbol{Z}^{k-1}) \approx \mathcal{IW}(X_k; \nu_{k|k-1}, X_{k|k-1}), \tag{4.35}$$

where the predicted parameter $\nu_{k|k-1}$ and $V_{k|k-1}$ are given in Table 2, which presents the pseudo code of the prediction step in an algorithm using random matrices.

Table 2: The Random Matrix prediction step

```
Estimated kinematics state and covariance matrix:
1:  $\hat{\boldsymbol{x}}_{k|k-1} = (F_{k|k-1} \otimes I_d)\, \hat{\boldsymbol{x}}_{k-1|k-1}$
2:  $P_{k|k-1} = F_{k|k-1} P_{k-1|k-1} F_{k|k-1}^T + Q_{k|k-1}$
Degrees of freedom and inverse scale matrix:
3:  $\nu_{k|k-1} = \nu_{k-1|k-1} e^{-\frac{T_s}{\tau}}$
4:  $X_{k|k-1} = \frac{\nu_{k|k-1} - 1 - d}{\nu_{k-1|k-1} - 1 - d} X_{k-1|k-1}$
```

Note that the calculation in Table 2 of the degree of freedom (line 3) and the inverse scale matrix (line 4) are heuristic approaches proposed by Koch [7, p. 1046]. It is assumed that the expectation values of the predicted density and the previous update density shall be equal, i.e. $\mathbb{E}\{X_{k|k-1}\} = \mathbb{E}\{X_{k-1|k-1}\}$[7] and the parameter $\nu_{k-1|k-1}$ should decrease using some time dependent damping described through $e^{-\frac{T_s}{\tau}}$, with τ as a decoy constant. τ describes the agility of the object's extension change. Higher τ means a less agile change of the object.
Alternative prediction methods that use different heuristics or that can incorporate nonlinear models (e.g. prediction of the turn rate) are summarised in [35].

4.3.4. The random matrix update step

The Bayesian update step was defined in (4.28) and the measurement model of a single object generated measurement is given in (4.15). The sensor model that is used in Koch's approach implies that the single measurements are spread over the object's extent as a Gaussian distribution. Further, it is assumed that the senor noise itself is negligible relative to the object's extent. These facts become obvious when the sensor specific likelihood $p(\boldsymbol{Z}_k, n_{\boldsymbol{Z}_k}|\boldsymbol{x}_k, X_k)$ in (4.28) is rearranged with the law of conditional probability, thus

$$p(\boldsymbol{Z}_k, n_{\boldsymbol{Z}_k}|\boldsymbol{x}_k, X_k) = p(\boldsymbol{Z}_k|n_{\boldsymbol{Z}_k}, \boldsymbol{x}_k, X_k) p(n_{\boldsymbol{Z}_k}|\boldsymbol{x}_k, X_k). \tag{4.36}$$

Now the conditional density $p(\boldsymbol{Z}_k|n_{\boldsymbol{Z}_k}, \boldsymbol{x}_k, X_k)$ can be factorized using the Gaussian product theorem, see [7], given as

$$p(\boldsymbol{Z}_k|n_{\boldsymbol{Z}_k}, \boldsymbol{x}_k, X_k) = \prod_{j=1}^{n_{Z_k}} \mathcal{N}\left(\boldsymbol{z}_k^{(j)}; (H_k \otimes I_d)\boldsymbol{x}_k, X_k\right) \tag{4.37}$$

[7] The expectation of the Wishart density is given in the Appendix A.3.

It can be seen that the measurement spread mainly arises from the object's extent X_k that is the assumed covariance matrix in the Gaussian density. The density for the number of measurements $p(n_{\mathbf{Z}_k}|\boldsymbol{x}_k, X_k)$ is assumed to be constant, i.e. independent of $\boldsymbol{x}_k$ and X_k. Further, for simplification clutter measurements are neglected. Note that further modelling of $p(n_{\mathbf{Z}_k}|\boldsymbol{x}_k, X_k)$ is described in the context of the GIW-PHD filter in chapter 5.

The density in (4.37) is further proportional to a product of Gaussian and Wishart distribution given as [7]

$$p(\boldsymbol{Z}_k|n_{\mathbf{Z}_k}, \boldsymbol{x}_k, X_k) \propto \mathcal{N}(\bar{\mathbf{z}}_k; (H_k \otimes I_d)\boldsymbol{x}_k, \frac{X_k}{n_{\mathbf{Z}_k}}) \cdot \mathcal{W}(\bar{Z}_k; n_{\mathbf{Z}_k} - 1, X_k), \tag{4.38}$$

where two statistical factors, the centroid measurement $\bar{\mathbf{z}}_k$ and scatter matrix $\bar{Z}_k$ are given in Table 3.

With the product of (4.31), (4.35) and (4.38) the Bayes update formula in (4.28) can be applied, which leads to the update step [35], i.e. the density after filtering in form of

$$p(\boldsymbol{x}_k, X_k|\boldsymbol{Z}^k) = \mathcal{N}(\boldsymbol{x}_k; \boldsymbol{x}_{k|k}, P_{k|k} \otimes X_k)\mathcal{IW}(X_k; \nu_{k|k}, X_{k|k}), \tag{4.39}$$

where the pseudo code for calculation the components of the densities is given in Table 3:

Table 3: The Random Matrix update step

```
Preparation of update components
$\bar{\mathbf{z}}_k = \frac{1}{n_{\mathbf{Z}_k}} \sum_{j=1}^{n_{\mathbf{Z}_k}} \mathbf{z}_k^{(j)}$
$\bar{Z}_k = \sum_{j=1}^{n_{\mathbf{Z}_k}} \left(\mathbf{z}_k^{(j)} - \bar{\mathbf{z}}_k\right)\left(\mathbf{z}_k^{(j)} - \bar{\mathbf{z}}_k\right)^T$
$\varepsilon_{k|k-1} = \bar{\mathbf{z}}_k - (H_k \otimes I_d)\hat{\boldsymbol{x}}_{k|k-1}$
$S_{k|k-1} = H_k P_{k|k-1} (H_k)^T + \frac{1}{n_{\mathbf{Z}_k}}$
$K_{k|k-1} = P_{k|k-1} (H_k)^T S_{k|k-1}^{-1}$
$N_{k|k-1} = S_{k|k-1}^{-1} \varepsilon_{k|k-1} (\varepsilon_{k|k-1})^T$
Computation of updated components
$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + (K_{k|k-1} \otimes I_d)\varepsilon_{k|k-1}$
$P_{k|k} = P_{k|k-1} - K_{k|k-1} S_{k|k-1} (K_{k|k-1})^T$
$\nu_{k|k} = \nu_{k|k-1} + n_{\mathbf{Z}_k}$
$X_{k|k} = X_{k|k-1} + N_{k|k-1} + \bar{Z}_k$
```

Note that alternative update methods that explicitly incorporate sensor errors are proposed in [40] and [41]. There, the measurement model $p(\boldsymbol{Z}_k|n_{\mathbf{Z}_k}, \boldsymbol{x}_k, X_k)$ is adopted in such a way that the covariance of the measurement likelihood $\tilde{R}_k$ is either the sum of extent matrix X_k and a sensor error covariance R with $\tilde{R}_k = X_k + R$ [41] or even the sum of a scaled extent matrix zX_k plus a sensor error covariance R with $\tilde{R}_k = zX_k + R$ [40]. Both variants adapt the measurement spread as $p(\boldsymbol{Z}_k|n_{\mathbf{Z}_k}, \boldsymbol{x}_k, X_k) = \prod_{j=1}^{n_{\mathbf{z}_k}} \mathcal{N}\left(\mathbf{z}_k^{(j)}; (H_k \otimes I_d)\boldsymbol{x}_k, \tilde{R}_k\right)$. These variants allow to model the sensor error with covariance R and with the scaling factor z, which makes it possible to spread the normal distribution. This is preferable in real applications when the measurement spread is rather uniform distributed and thus a scaled Gaussian can be seen as an approximation to it [35, p. 9]. For further update schemes see the comprehensive summery of extended object tracking in [35].

5. Multiple extended object tracking

In this chapter, the multiple extended object tracking problem is treated with the presentation of the Gaussian Inverse Wishart PHD filter [6]. This filter is used to investigate multi-object scenarios. The filter combines the PHD filter for extended objects, which was derived by Mahler in [8], with the random matrix framework of Koch [7] presented in chapter 4. The resulting filter is applicable in scenarios with missed detections, clutter and a known measurement rate of elliptical shaped objects. In section 5.1, a short capture is given to extended objects in PHD filters, whereas the literature of Mahler's derivation is mentioned as well as precursor works of Granström with the Gaussian mixture implementations. In section 5.2, the multiple extended object tracking problem is defined with linkage to the previous chapters. Finally in section 5.3, the GIW-PHD filter of [6] is presented with necessary assumptions, prediction and correction equations. In the end of the chapter, the filter procedure is summarized and the essential pruning and merging scheme is given.

5.1. Extended objects in PHD filters

The multi-object tracking problem was discussed in chapter 3, where traditional data association methods combined with single point object Bayes filter and the RFS set approach were addressed. Further, a practical implementation of an RFS approach was given with the PHD filter for point objects of Vo et al. [12] in section 3.5. For multi-object scenarios where the point object assumption does not hold anymore, a filter approach that can process multiple measurements per object becomes necessary. An extended object model is given by Gilholm et al. [36] with the spatial Poisson model. This extended object model is captured by Mahler and his derivation of the PHD filter for extended objects in [8], where he assumes the measurement likelihood function to be a Poisson process of form

$$h_{k|k}\left(\boldsymbol{Z}_k\middle|\boldsymbol{x}_k^{(i)}\right) = L_{\boldsymbol{Z}_k}\left(\boldsymbol{x}_k^{(i)}\right) = e^{-\gamma\left(\boldsymbol{x}_k^{(i)}\right)} \prod_{\boldsymbol{z}\in\boldsymbol{Z}_k} \gamma\left(\boldsymbol{x}_k^{(i)}\right) \Phi_{\boldsymbol{z}}(\boldsymbol{x}_k^{(i)}), \tag{5.1}$$

where $\gamma\left(\boldsymbol{x}_k^{(i)}\right)$ is the expected number of measurements and $\Phi_{\boldsymbol{z}}\left(\boldsymbol{x}_k^{(i)}\right) = \Phi_k\left(\boldsymbol{z}_k\middle|\boldsymbol{x}_k^{(i)}\right)$ is the spatial distribution of a single measurement. With this measurement model Mahler derived the exact measurement update equations for the extended object PHD filter. Granström combined the extended object PHD filter with the random matrix framework [7], i.e. he assumed the extended objects to be elliptically shaped which results the spatial distribution $\Phi_k(\cdot\mid\cdot)$ to be a product of Gaussian distributions as shown in (4.37). The resulting filter is the GIW-PHD filter, presented in [6], that approximates the PHD with a mixture of GIW components.

In early works of Granström, [9] and [42], an extended target GM-PHD filter was presented, where only the kinematic properties of the objects' centroids were estimated. Modelling the objects' extent was omitted but is explicitly done in [6] with the GIW-PHD implementation, which is presented in the following sections.

5.2. The multiple extended object tracking problem

Let the set of extended objects ξ_k at time step k be

$$\boldsymbol{X}_k = \left\{\xi_k^{(i)}\right\}_{i=1}^{n_{X_k}}, \tag{5.2}$$

where $n_{\boldsymbol{X}_k}$ is the unknown number of present objects and $\xi_k^{(i)}$ is the extended state vector mentioned in section 4.1. The operation $|\cdot|$ denotes the set cardinality, thus $|\boldsymbol{X}_k| = n_{\boldsymbol{X}_k}$. The extended state vector is an augmented state consisting of the kinematics state $\boldsymbol{x}_k^{(i)}$ and extension state $X_k^{(i)}$ of the ith object defined as

$$\xi_k^{(i)} \triangleq \left(\boldsymbol{x}_k^{(i)}, X_k^{(i)}\right). \tag{5.3}$$

$X_k^{(i)}$ represents the random matrix of the elliptical shaped objects. The objects' dynamic motion model used in the GIW-PHD filter is defined in section 4.3.1 according to [7].
The set of measurements that is obtained at time step k is denoted as

$$\boldsymbol{Z}_k = \left\{\boldsymbol{z}_k^{(j)}\right\}_{j=1}^{n_{Z_k}}, \tag{5.4}$$

where $n_{\boldsymbol{Z}_k} = |\boldsymbol{Z}_k|$ is the number of measurements. The measurement model according to [7] is given in section 4.3.1. Each object generates a Poisson distributed number of measurements with rate $\gamma\left(\xi_k^{(i)}\right)$ dependent on the augmented object state. Further, the presence of clutter is modelled, i.e. the number of clutter measurements per time step k is Poisson distributed with the rate parameter $\beta_{FA,k}$ that determines the clutter measurements (false alarms) per surveillance volume $\mathcal{S}$ per time step. Therefore, the mean number of clutter measurements states

$$\lambda_{FA,k} = \beta_{FA,k}\mathcal{S}, \tag{5.5}$$

where $\mathcal{S}$ is of dimension d. In this thesis the surveillance volume is an area in 2-dimensional space in Cartesian coordinates. The clutter measurements in this area are modelled as uniformly distributed.
The goal of each time step is to create an estimate of the extended object state that is denoted as $\hat{\xi}_{k|k}^{(i)}$ Therefore, the object states $\boldsymbol{X}_k$ and the measurement sequence $\boldsymbol{Z}^k = \{\boldsymbol{Z}_m\}_{m=1}^k$ is processed in a Bayesian framework as a distribution $p(\boldsymbol{X}_k|\boldsymbol{Z}^k)$ that is approximated with the PHD $D_{k|k}$ and its propagation through time, cf. (3.39).

5.3. The Gaussian inverse Wishart PHD Filter

For the multi-object tracking problem, the PHD filter equations of the prediction step depicted in (3.40) are adapted here to the extended state vector ξ_k and without object spawning, given as

$$\begin{aligned} D_{k|k-1}(\xi_k) &= D_{\gamma,k-1}(\xi_k) \\ &\quad + \int p_S(\xi_{k-1}) f_{k|k-1}(\xi_k|\xi_{k-1})\, D_{k-1|\mathrm{k}-1}(\xi_{k-1}) d\xi_{k-1}, \end{aligned} \tag{5.6}$$

where

- $D_{\gamma,k-1}(\cdot)$ is the birth PHD that represents new objects in defined areas of the surveillance volume $\mathcal{S}$,
- $p_S(\cdot)$ is the probability of survival as function of the augmented object state,
- $f_{k|k-1}(\cdot\,|\,\cdot)$ is the state transition density that describes the object evolution from ξ_{k-1} to ξ_k.

In [8] the correction equations for the PHD filter with extended objects are given as the product of the sensor pseudo likelihood function and the predicted PHD in following form

$$D_{k|k}(\xi_k|\boldsymbol{Z}^k) = L_{\boldsymbol{Z}_k}(\xi_k) D_{k|k-1}(\xi_k|\boldsymbol{Z}^{k-1}). \tag{5.7}$$

The measurement pseudo likelihood function $L_{\boldsymbol{Z}_k}(\cdot)$ in (5.7) is derived in [8] and [3]:

For the non-detection case of an object, i.e. $\boldsymbol{Z}_k = \emptyset$, it is given as

$$L_{\boldsymbol{Z}_k}(\xi_k) = 1 - p_{D,k}(\xi_k) + e^{-\gamma(\xi_k)} p_{D,k}(\xi_k) \tag{5.8}$$

and for the detection case of an object as

$$\begin{aligned} L_{\boldsymbol{Z}_k}(\xi_k) = 1 - p_{D,k}(\xi_k) + e^{-\gamma(\xi_k)} p_{D,k}(\xi_k) + e^{-\gamma(\xi_k)} p_{D,k}(\xi_k) \\ \cdot \sum_{\wp \angle \boldsymbol{Z}_k} \omega_{\mathcal{P}} \sum_{\boldsymbol{W} \in \wp} \frac{\gamma(\xi_k)^{|W|}}{d_{\boldsymbol{W}}} \prod_{\boldsymbol{z} \in W} \frac{\phi_{\boldsymbol{z}}(\xi_k)}{\lambda_{FA,k} c_k(\boldsymbol{z})} \end{aligned} \tag{5.9}$$

where

- $\lambda_{FA,k}$ is the mean of clutter, cf. (5.5),
- $c_k(\boldsymbol{z})$ is the spatial distribution of the clutter measurements, uniformly distributed over the surveillance area, i.e. $c_k(\boldsymbol{z}) = 1/\mathcal{S}$,
- further the abbreviation $\wp \angle \boldsymbol{Z}_k$ under the first summation sign in (5.9) means that the sum is taken over all possible partitions $\wp$ of the partition set $\mathcal{P}$, see Table 4,
- the abbreviation $\boldsymbol{W} \in \wp$ under the second summation sign in (5.9) denotes that the sum is taken over all cells $\boldsymbol{W}$ in the current partition $\wp$ of $\mathcal{P}$, see Table 4.

Table 4: Excursion - partitions and cells

A measurement set $\boldsymbol{Z}$ of single measurements $\boldsymbol{z}^{(j)}$ can be divided into groups of measurements that are called cells $\boldsymbol{W}$ or sometimes clusters as well. It is possible to build a cell $\boldsymbol{W}$ with at least one or more measurements $\boldsymbol{z}^{(j)}$. The union of all cells $\boldsymbol{W}_i$, where i is the cell index, must contain all measurements in $\boldsymbol{Z}$ and is then called partition $\wp$, i.e. $\cup_{W_i \in \wp} = \boldsymbol{Z}$. Due to the fact that every cell can be built with different measurements, there arise several ways to build a partition $\wp$. The set of all partitions is called partition set $\mathcal{P} = \{\wp^{(1)}, \dots, \wp^{(p)}\}$.
Example: Let the number of measurements be $|\boldsymbol{Z}| = 3$ and the set $\boldsymbol{Z} = \{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \boldsymbol{z}^{(3)}\}$. The number of all possible partitions p of $\boldsymbol{Z}$ is determined by the Bell' number, thus $p = bell(3) = 5$, where the partition set $\mathcal{P}$ consist of following single partitions $\wp^{(p)}$:

$$\begin{aligned} \wp^{(1)} &= \{\boldsymbol{W}_1^{(1)}\} = \{\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \boldsymbol{z}^{(3)}\}\} \\ \wp^{(2)} &= \{\boldsymbol{W}_1^{(2)}, \boldsymbol{W}_2^{(2)}\} = \{\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}\}, \{\boldsymbol{z}^{(3)}\}\} \\ \wp^{(3)} &= \{\boldsymbol{W}_1^{(3)}, \boldsymbol{W}_2^{(3)}\} = \{\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(3)}\}, \{\boldsymbol{z}^{(2)}\}\} \\ \wp^{(4)} &= \{\boldsymbol{W}_1^{(4)}, \boldsymbol{W}_2^{(4)}\} = \{\{\boldsymbol{z}^{(2)}, \boldsymbol{z}^{(3)}\}, \{\boldsymbol{z}^{(1)}\}\} \\ \wp^{(5)} &= \{\boldsymbol{W}_1^{(5)}, \boldsymbol{W}_2^{(5)}, \boldsymbol{W}_3^{(5)}\} = \{\{\boldsymbol{z}^{(1)}\}, \{\boldsymbol{z}^{(2)}\}, \{\boldsymbol{z}^{(3)}\}\}. \end{aligned}$$

It can be seen that the partition $\wp^{(1)}$ consists of one cell $\boldsymbol{W}_i^{(1)}, i = 1$ that contains all measurements. The partitions $\wp^{(2)}, \wp^{(3)}, \wp^{(4)}$ have one cell with two measurements and one cell with one measurement. In partition $\wp^{(5)}$ every measurement is placed in a single cell $\boldsymbol{W}_i^{(5)}, i = 1,2,3$. More information about partitioning and methods is given in chapter 6.

In (5.9) there are two normalization coefficients $\omega_\wp$ and $d_{\boldsymbol{W}}$. The coefficient $\omega_\wp$ is the "relative weight" of a partition $\wp$ where $\sum_{\wp \in \mathcal{P}} \omega_\wp = 1$. It is defined as

$$\omega_\wp = \frac{\prod_{W \in \wp} d_W}{\sum_{\wp' \angle Z_k} \prod_{W \in \wp'} d_W}. \tag{5.10}$$

The normalization factor $d_{\boldsymbol{W}}$ or "cell weight" of a single cell $\boldsymbol{W}$ contains two summands, the Kronecker delta and the value of the predicted PHD of all measurements in cell $\boldsymbol{W}$. $d_{\boldsymbol{W}}$ is defined as

$$d_{\boldsymbol{W}} = \delta_{|\boldsymbol{W}|,1} + D_{k|k-1}\left[p_D \gamma^{|\boldsymbol{W}|} e^{-\gamma} \prod_{\mathbf{z}\in\boldsymbol{W}} \frac{\Phi_{\mathbf{z}}(\xi_k)}{\lambda_{FA,k} c_k(\mathbf{z})}\right]. \tag{5.11}$$

In (5.11) $\delta_{n,m}$ denotes the Kronecker delta[8] and the operation $f[g]$ denotes the integral $\int f(x)g(x)dx$.

Now it is assumed that the PHD $D_{k|k}(\cdot)$ at time step k and for all following time steps, is an unnormalised mixture of Gaussian inverse Wishart densities [6] on the single object state space $\mathcal{X}_0$ given

$$D_{k|k}(\xi_k) \approx \sum_{j=1}^{J_{k|k}} w_{k|k}^{(j)}\ \mathcal{N}\left(\boldsymbol{x}_k; \boldsymbol{m}_{k|k}^{(j)}, P_{k|k}^{(j)} \otimes X_k\right) \cdot \mathcal{IW}\left(X_k; v_{k|k}^{(j)}, V_{k|k}^{(j)}\right). \tag{5.12}$$

The parameters in (5.12) are

- $J_{k|k}$ the number of components of the GIW mixture,
- $w_{k|k}^{(j)}$ the weight of the jth component,
- $\boldsymbol{m}_{k|k}^{(j)}$ and $P_{k|k}^{(j)} \otimes X_k$ as the respective expectation and covariance matrix of the jth component,
- $v_{k|k}^{(j)}$ the degrees of freedom of the jth component of the inverse Wishart distribution,
- $V_{k|k}^{(j)}$ the inverse scale matrix of the jth component.

The augmented state vector $\xi_{(\cdot)}^{(j)}$ with index (j) is an abbreviation that contains the parameter of both the Gaussian and Inverse Wishart distribution in (5.12) of the jth component:

$$\xi_{k|k}^{(j)} = \left(\boldsymbol{m}_{k|k}^{(j)}, P_{k|k}^{(j)}, v_{k|k}^{(j)}, V_{k|k}^{(j)}\right) \tag{5.13}$$

An estimate of the object's extent $\hat{X}_{k|k}$ and the kinematics uncertainty $\hat{P}_{k|k}$ can be obtained as in [7] with

$$\hat{X}_{k|k}^{(j)} = \frac{V_{k|k}^{(j)}}{v_{k|k}^{(j)} - 2d - 2} \tag{5.14}$$

and

$$\hat{P}_{k|k}^{(j)} = \frac{P_{k|k}^{(j)} \otimes V_{k|k}^{(j)}}{v_{k|k}^{(j)} + s - sd - 2}. \tag{5.15}$$

5.3.1. Filter assumptions

For the derivation of the filter equations of prediction and correction the following assumptions are made in [6]:

A.1. Each object is independent of all other objects regarding its motion and measurement generation process.

A.2. The object's kinematics part follows a linear Gaussian motion model and the sensor follows a linear Gaussian measurement model, see section 4.3.1.

[8] The Kronecker delta definition is given in Appendix A.2.

A.3. Clutter or false alarm measurements are Poisson distributed in number with rate $\gamma(\xi_k)$ and independent of object-originated measurements.

A.4. The survival probability of each object is state independent, thus $p_S(\xi_k) = p_{S,k}$.

Assumptions A.1 - A.4 are standard in most object tracking applications, see e.g. [1]. The next assumption can be assumed in applications if object interactions, e.g. merging or spawning, is negligible [12]:

A.5. The predicted multi-object RFS is Poisson.

A.6. The birth PHD, cf. (5.6), is a mixture of GIW distributions.

A.7. The object augmented state transition density can be split into a product of the kinematic evolution and the extent evolution, shown in (4.24), as

$$f_{k|k-1}(\xi_k|\xi_{k|k-1}) \approx \underbrace{p(\boldsymbol{x}_k|X_k,\boldsymbol{x}_{k-1})}_{kinematic\ evolution} \underbrace{p(X_k|X_{k-1})}_{extent\ evolution} . \tag{5.16}$$

A.8. The following approximation about the detection probability $p_D(\cdot)$ holds for all states ξ_k:

$$\begin{aligned} &p_D(\xi_k)\mathcal{N}(\boldsymbol{x}_k;\boldsymbol{m}_{k|k-1},P_{k|k-1}\otimes X_k)\cdot\mathcal{IW}(X_k;v_{k|k-1},V_{k|k-1}) \\ &\approx p_D\left(\xi_{k|k-1}^{(j)}\right)\mathcal{N}(\boldsymbol{x}_k;\boldsymbol{m}_{k|k-1},P_{k|k-1}\otimes X_k)\cdot\mathcal{IW}(X_k;v_{k|k-1},V_{k|k-1}). \end{aligned} \tag{5.17}$$

Let $p_D^{(j)} \triangleq p_D(\xi_{k|k-1}^{(j)})$ be an abbreviation of the detection probability for the jth GIW component in the remainder of this thesis. In this thesis (5.17) is satisfied, because $p_D(\cdot$ $)$ is set constant, i.e. $p_D(\cdot) = p_D$.

A.9. The approximation for the expected number of measurements, represented by rate $\gamma(\cdot)$, holds for all ξ_k:

$$\begin{aligned} &e^{-\gamma(\xi_k)}\gamma^n(\xi_k)\mathcal{N}(\boldsymbol{x}_k;\boldsymbol{m}_{k|k-1},P_{k|k-1}\otimes X_k)\cdot\mathcal{IW}(X_k;v_{k|k-1},V_{k|k-1}) \\ &\approx e^{-\gamma\left(\xi_{k|k-1}^{(j)}\right)}\gamma^n\left(\xi_{k|k-1}^{(j)}\right)\mathcal{N}(\boldsymbol{x}_k;\boldsymbol{m}_{k|k-1},P_{k|k-1}\otimes X_k) \\ &\qquad\cdot\mathcal{IW}(X_k;v_{k|k-1},V_{k|k-1}). \end{aligned} \tag{5.18}$$

Let $\gamma^{(j)} \triangleq \gamma(\xi_{k|k-1})$ be an abbreviation for the expected number of measurements for the jth GIW component in the remainder of this thesis. In this thesis (5.18) is satisfied, because $\gamma(\cdot)$ is set constant, i.e. $\gamma(\cdot) = \gamma$.

For further discussion of assumptions A.8 and A.9 see [6, pp. 4-5]. With these assumptions the prediction and correction step are presented in the next two subsections.

5.3.2. Prediction step

According to the assumptions A.4 and A.7, the prediction of existing objects is performed as in the random matrix framework, see section 4.3.3. The integral in (4.24) is extended with the constant survival probability p_S and a sum over the single components with indices j. Then, the prediction density is stated as

$$\begin{aligned} p(\xi_k|\boldsymbol{Z}^{k-1}) = p_S \sum_{j=1}^{J_{k-1|k-1}} w_{k-1|k-1}^{(j)} &\underbrace{\int p(\boldsymbol{x}_k|X_k,\boldsymbol{x}_{k-1})\,p(\boldsymbol{x}_{k-1}|X_k,\boldsymbol{Z}^{k-1})d\boldsymbol{x}_{k-1}}_{kinematic\ part} \\ &\cdot \underbrace{\int p(X_k|X_{k-1})p(X_{k-1}|\boldsymbol{Z}^{k-1})dX_{k-1}}_{extension\ part} \end{aligned} \tag{5.19}$$

When inserting the densities for the kinematics part (4.31) and the extension part (4.35), the PHD that corresponds to the prediction of existing object becomes [6, p. 5]

$$\sum_{j=1}^{J_{k-1|k-1}} w_{k|k-1}^{(j)} \mathcal{N}\left(\boldsymbol{x}_k; \boldsymbol{m}_{k|k-1}^{(j)}, P_{k|k-1}^{(j)} \otimes X_k\right) \cdot \mathcal{IW}\left(X_k; \nu_{k|k-1}^{(j)}, V_{k|k-1}^{(j)}\right), \tag{5.20}$$

where the calculations of the components in (5.20) are given in Table 5.

The birth PHD $D_{k-1}^{\gamma}(\xi_k)$ of (5.6), which represents new objects appearing at time $k-1$, is given as

$$D_{k-1}^{\gamma}(\xi_k) = \sum_{j=1}^{J_{\gamma,k-1}} w_{\gamma,k-1}^{(j)} \mathcal{N}\left(\boldsymbol{x}_k; \boldsymbol{m}_{\gamma,k-1}^{(j)}, P_{\gamma,k-1}^{(j)} \otimes X_k\right) \cdot \mathcal{IW}\left(X_k; \nu_{\gamma,k-1}^{(j)}, V_{\gamma,k-1}^{(j)}\right) \tag{5.21}$$

In total, the predicted PHD $D_{k|k-1}(\xi_k)$ that is stated in (5.6) has $J_{k|k-1} = J_{k-1|k-1} + J_{\gamma,k-1}$ GIW mixture components.

Table 5: The GIW-PHD filter prediction components

Predicted kinematics mean and covariance matrix:

1: $\boldsymbol{m}_{k|k-1}^{(j)} = (F_{k|k-1} \otimes I_d)\, \boldsymbol{m}_{k-1|k-1}^{(j)}$

2: $P_{k|k-1}^{(j)} = F_{k|k-1} P_{k-1|k-1}^{(j)} F_{k|k-1}^{T} + Q_{k|k-1}$

Predicted degrees of freedom and inverse scale matrix:

3: $\nu_{k|k-1}^{(j)} = \nu_{k-1|k-1}^{(j)} e^{-\frac{T_S}{\tau}}$[9]

4: $V_{k|k-1}^{(j)} = \frac{\nu_{k|k-1}^{(j)} - 1 - d}{\nu_{k-1|k-1}^{(j)} - 1 - d} V_{k-1|k-1}^{(j)}$

Predicted weight of GIW component

5: $w_{k|k-1}^{(j)} = p_S w_{k-1|k-1}^{(j)}$

5.3.3. Correction step

The corrected PHD $D_{k|k}(\cdot)$ is given as the sum of the not detected and detected PHD as [6]

$$D_{k|k}(\xi_k) = D_{k|k}^{ND}(\xi_k) + \sum_{\wp \angle \boldsymbol{Z}_k} \sum_{\boldsymbol{W} \in \wp} D_{k|k}^{D}(\xi_k, \boldsymbol{W}) \tag{5.22}$$

Due to the probability of detection $p_D < 1$ it could happen that objects are not detected. Then the PHD $D_{k|k}^{ND}(\cdot)$ in form of a GIW mixture is given by

$$D_{k|k}^{ND}(\xi_k) = \sum_{j=1}^{J_{k|k-1}} w_{k|k}^{(j)} \mathcal{N}\left(\boldsymbol{x}_k; \boldsymbol{m}_{k|k}^{(j)}, P_{k|k}^{(j)} \otimes X_k\right) \mathcal{IW}\left(X_k; \nu_{k|k}^{(j)}, V_{k|k}^{(j)}\right), \tag{5.23}$$

where the updated components of the Gaussian and inverse Wishart distribution are carried over unchanged as

$$\xi_{k|k}^{(j)} = \xi_{k|k-1}^{(j)}. \tag{5.24}$$

The weights of the non-detection case in (5.24) are decreased as

$$w_{k|k}^{(j)} = \left(1 - \left(1 - e^{-\gamma(j)}\right) p_D\right) \cdot w_{k|k-1}^{(j)}, \tag{5.25}$$

[9] τ see Table 2.

where the factor $\left(1 - e^{-\gamma^{(j)}}\right) p_D$ denotes the effective probability of detection [9] and thus $\left(1 - \left(1 - e^{-\gamma^{(j)}}\right) p_D\right)$ is the complementary event of non-detection. The effective probability of detection consists of the product of $\left(1 - e^{-\gamma^{(j)}}\right)$, as the resulting probability of a Poisson distribution to generate at least one measurement, and p_D, as the assumed constant sensor detection probability.

The objects that are detected are captured in the PHD $D_{k|k}^{D}(\xi_k, \boldsymbol{W})$, which is a product of sensor likelihood function and the predicted PHD, cf. (5.7). So the sensor likelihood in each cell $\boldsymbol{W}$,

$$\prod_{\boldsymbol{z}_k \in \boldsymbol{W}} \frac{\Phi_{\boldsymbol{z}_k}(\xi_k)}{\lambda_{FA,k} c_k(\mathbf{z}_\mathrm{k})} = \beta_{FA,k}^{-|\boldsymbol{W}|} \prod_{\boldsymbol{z}_k \in \boldsymbol{W}} \mathcal{N}\left(\boldsymbol{z}_k^{(i)}; (H_k \otimes I_d)\boldsymbol{x}_k, X_k\right), \tag{5.26}$$

multiplied with the predicted PHD components in (5.20)

$$\mathcal{N}\left(\boldsymbol{x}_k; \boldsymbol{m}_{k|k-1}^{(j)}, P_{k|k-1}^{(j)} \otimes X_k\right) \mathcal{IW}\left(X_k; v_{k|k-1}^{(j)}, V_{k|k-1}^{(j)}\right), \tag{5.27}$$

can be rewritten as [6]

$$\beta_{FA,k}^{-|W|} \mathcal{L}_k^{(j,W)} \mathcal{N}\left(\boldsymbol{x}_k; \boldsymbol{m}_{k|k}^{(j)}, P_{k|k}^{(j)} \otimes X_k\right) \mathcal{IW}\left(X_k; v_{k|k}^{(j)}, V_{k|k}^{(j)}\right). \tag{5.28}$$

Note that a detailed derivation of (5.28), which is a product of (5.25) and (5.26), can be found in the appendix of [6]. The corrected Gaussian and inverse Wishart components in (5.28) as well as the likelihood $\mathcal{L}_k^{(\cdot)}$ are given in Table 6. This correction is based on the random matrix framework update, cf. Table 3.

Table 6: The GIW-PHD filter correction components

Corrected Gaussian mean $\boldsymbol{m}$ and covariance P:

1: $\boldsymbol{m}_{k|k}^{(j,\boldsymbol{W})} = \boldsymbol{m}_{k|k-1}^{(j)} \left(K_{k|k-1}^{(j,\boldsymbol{W})} \otimes I_d\right) \varepsilon_{k|k-1}^{(j,\boldsymbol{W})}$

2: $P_{k|k}^{(j,\boldsymbol{W})} = P_{k|k-1}^{(j)} - K_{k|k-1}^{(j,\boldsymbol{W})} S_{k|k-1}^{(j,\boldsymbol{W})} \left(K_{k|k-1}^{(j,\boldsymbol{W})}\right)^T$

Corrected inverse Wishart degrees of freedom ν and inverse scale matrix V:

3: $\nu_{k|k}^{(j,\boldsymbol{W})} = \nu_{k|k-1}^{(j)} |\boldsymbol{W}|$

4: $V_{k|k}^{(j,\boldsymbol{W})} = V_{k|k-1}^{(j)} + N_{k|k-1}^{(j,\boldsymbol{W})} + \bar{Z}_k^{\boldsymbol{W}}$

Corrected weight w

5: $w_{k|k}^{(j,\boldsymbol{W})} = \frac{\omega_{\wp}}{d_{\boldsymbol{W}}} e^{-\gamma(j)} \left(\frac{\gamma(j)}{\beta_{FA,k}}\right)^{|\boldsymbol{W}|} p_D^{(j)} \mathcal{L}_k^{(j,\boldsymbol{W})} w_{k|k-1}^{(j)}$

with likelihood $\mathcal{L}$ as

6: $\mathcal{L}_k^{(j,\boldsymbol{W})} = \frac{1}{\left(\pi^{|\boldsymbol{W}|} |\boldsymbol{W}| S_{k|k-1}^{(j,\boldsymbol{W})}\right)^{d/2}} \cdot \frac{\left|V_{k|k-1}^{(j)}\right|^{\nu_{k|k-1}^{(j)}/2}}{\left|V_{k|k}^{(j,\boldsymbol{W})}\right|^{\nu_{k|k}^{(j,\boldsymbol{W})}/2}} \cdot \frac{\boldsymbol{\Gamma}_d(\nu_{k|k}^{(j,\boldsymbol{W})}/2)}{\boldsymbol{\Gamma}_d(\nu_{k|k-1}^{(j)}/2)}$, for $\boldsymbol{\Gamma}_d$ see Appendix A.3.2.,

normalization coefficients $\omega_{\wp}$, see (5.10)

7: $d_{\boldsymbol{W}} = \delta_{|\boldsymbol{W}|,1} + \sum_{l=1}^{J_{k|k-1}} e^{-\gamma(l)} \left(\frac{\gamma(l)}{\beta_{FA,k}}\right)^{|\boldsymbol{W}|} p_D^{(l)} \mathcal{L}_k^{(l,\boldsymbol{W})} w_{k|k-1}^{(l)}$.

Centroid measurement $\bar{\mathbf{z}}$ and scatter matrix $\bar{Z}$:

8: $\bar{\boldsymbol{z}}_k^{\boldsymbol{W}} = \frac{1}{|\boldsymbol{W}|} \sum_{\boldsymbol{z}_k^{(i)} \in \boldsymbol{W}} \boldsymbol{z}_k^{(i)}$

9: $\bar{Z}_k^{|\boldsymbol{W}|} = \sum_{\boldsymbol{z}_k^{(i)} \in \boldsymbol{W}} \left(\boldsymbol{z}_k^{(j)} - \bar{\boldsymbol{z}}_k^{\boldsymbol{W}}\right) \left(\boldsymbol{z}_k^{(j)} - \bar{\boldsymbol{z}}_k^{\boldsymbol{W}}\right)^T$

Innovation vector ε and innovation factor S:

10: $\varepsilon_{k|k-1}^{(j,\boldsymbol{W})} = \bar{\boldsymbol{z}}_k^{\boldsymbol{W}} - (H_k \otimes I_d) m_{k|k-1}^{(j)}$

```
$S_{k|k-1}^{(j,\boldsymbol{W})} = H_k P_{k|k-1}^{(j)} (H_k)^T + \frac{1}{|\boldsymbol{W}|}$
Gain matrix $K$ and innovation matrix $N$
$K_{k|k-1}^{(j,\boldsymbol{W})} = P_{k|k-1}^{(j)} (H_k)^T \left(S_{k|k-1}^{(j,\boldsymbol{W})}\right)^{-1}$
$N_{k|k-1}^{(j,\boldsymbol{W})} = \left(S_{k|k-1}^{(j,\boldsymbol{W})}\right)^{-1} \varepsilon_{k|k-1}^{(j,\boldsymbol{W})} \left(\varepsilon_{k|k-1}^{(j,\boldsymbol{W})}\right)^T$
```

Notes: The operation $|V|$ denotes the determinant of matrix V and $|\boldsymbol{W}|$ determines the number of measurements in cell $\boldsymbol{W}$.

Finally, the number of corrected PHD components is given as

$$J_{k|k} = J_{k|k-1} + J_{k|k-1} \sum_{m=1}^{p} |\wp^{(m)}| \, , \tag{5.29}$$

whereas $|\wp^{(m)}|$ denotes the number of cells $\boldsymbol{W}$ in the mth partition.

5.3.4. Implementation aspects

Through the prediction and correction step in the filter the number of filter components $J_{k|k}$ increases rapidly over time. To handle this, the pruning and merging of components have to be performed in a similar way as for the GM-PHD filter components, cf. Appendix A.1, Table 25. The adapted scheme for the GIW mixture components is given in the Appendix A.4, Table 27. Remark that for the merged covariance $\tilde{P}_{k|k}^{(l)}$ the spread of means is neglected. The spread of means $\left(\tilde{\boldsymbol{m}}_k^{(l)} - \boldsymbol{m}_k^{(i)}\right)\left(\tilde{\boldsymbol{m}}_k^{(l)} - \boldsymbol{m}_k^{(i)}\right)^T$ was added in the case of the GM-PHD filter to the merged covariance $\tilde{P}_{k|k}^{(l)}$, cf. Table 25. In the case of the GIW-PHD filter this term is neglected, cf. Table 27, because covariance $P_{k|k}^{(i)}$ and mean $\boldsymbol{m}_{k|k}^{(i)}$ are of different dimensions. The authors of [6] state that this neglect is justified because in the general case the spread of means is very small.

A crucial part of the correction step is the partitioning of the measurements into cells. This procedure is discussed in detail in chapter 6, where two classical partitioning methods are considered and a new approach that incorporates the multi-object likelihood is presented for the GIW-PHD filter. For the whole GIW-PHD filter a pseudo-code implementation is given in [43], where the single filter steps are presented in detail for implementation. An overview about the single steps in the filter procedure is given in Table 7.

Table 7: Classical GIW-PHD filter steps

```
**Given** Sequence of measurement sets $\{\boldsymbol{Z}_k\}_{k=1}^N$
**Initialize** $J_{0|0} = 0$
for $k = 1, \dots, N$
        Compute the partition set $\mathcal{P} = \left\{\wp^{(m)}\right\}_{m=1}^{p}$ of the measurements, see chapter 6
        Predict the GIW mixture components, see Table 5 or [43, p. 2]
        Correct the GIW mixture components, see Table 6 or [43, p. 3]
        Prune and merge the corrected components, see Table 27
        Extract the estimated objects, analogue to Table 26[10]
end for
**Output** sequence of estimated object sets $\left\{\widehat{\boldsymbol{X}}_k\right\}_{k=1}^N$ with GIW mixture components
```

[10] Replace the GM components $\left\{w_{k|k}^{(i)}, \boldsymbol{m}_{k|k}^{(i)}, P_{k|k}^{(i)}\right\}_{i=1}^{J_{k|k}}$ by the GIW mixture components $\left\{w_{k|k}^{(i)}, \xi_{k|k}^{(i)}\right\}_{i=1}^{J_{k|k}}$.

6. Likelihood-based measurement partitioning

In this chapter, the clustering of measurement sets and the data association problem is addressed, which is a crucial part in multi-object tracking, when objects are closely spaced. In traditional tracking algorithms this is done in two steps. First, partitioning of measurements using clustering algorithms to place measurements, which originate from the same object, in preferably the same cell. Second, for the several measurement cells the sources - in general objects – have to be associated. The classical approaches are presented in 6.1, where the proposed methods of [6] called Distance Partitioning and Sub-Partitioning are discussed. Additionally, some scientific work of the last few years is revealed, which combines some classical methods with add-on approaches to improve the filter performance.

On the other hand, new approaches, which incorporate the multi-object likelihood and therefore are more promising for closely spaced objects, are presented in section 6.2. The likelihood-based methods use stochastic sampling methods that randomly change the data association or partition. For the GIW-PHD filter, the new method called Stochastic Partitioning (StP) is proposed, which is based on the Stochastic Optimisation (SO) in [10].

6.1. Classical clustering and data association

In classical multi-object tracking scenarios, the objects have been treated as point objects, which give at most one measurement per time step. The measurement to track association in such cases is limited to the association of single measurements to the predicted object. Such methods of point object data association were discussed in chapter 3.

With the rise of multiple extended object tracking this part becomes more difficult because every object generates multiple measurements that have to be associated to predicted objects. The challenging fact is that the measurement origin is unknown and beside object generated measurements there are clutter measurements as well. This problem is classically solved in a 2-step approach of a previous measurement clustering followed by the cluster to object association. The measurement partitioning is done using clustering methods to find different ways in which the measurement set can be clustered. With the subsequent association using assignment methods, e.g. Auction algorithm [44] or Murty's algorithm [45], the different clusters are assigned to objects. The advantage of this 2-step approach is that the cluster can be assigned to objects using the point object assignment methods, which implies that each cluster is assigned to at most one object.

In the presented GIW-PHD filter in chapter 5 the measurement set $\boldsymbol{Z}$ is modelled as RFS and the filter algorithm needs for its correction step (5.22) all possible cells W of the partition set $\mathcal{P} = \{\wp^{(n)}\}_{n=1}^{p}$. Subsequently, the terms cell, partition and partition set are discussed.

A partition $\wp$ is a division of the measurement set $\boldsymbol{Z}$ into smaller groups, called cells or clusters. Cells are denoted in this thesis as $\boldsymbol{W}$. A requirement on a partition is that every measurement $\boldsymbol{z} \in \boldsymbol{Z}$ belongs to one and only one cell $\boldsymbol{W}$. With a clustering method the purpose is to obtain that the measurements of one object are clustered into the same cell $\boldsymbol{W}$. Now there are many ways to cluster a measurement set. Let a measurement set $\boldsymbol{Z}$ contain m measurements, $\boldsymbol{Z} = \{\boldsymbol{z}^{(i)}\}_{i=1}^{m}$, thus, the number of possible ways to partition $\boldsymbol{Z}$ is determined by the Bell number [46], denoted as $bell(n)$. For $n = 3$ measurements the number of possible partitions is $bell(3) = 5$, see Figure 2. For $n = 6$ measurements, there are $bell(6) = 203$ possible partitions and for $n = 90$ measurements the number of partitions is vast with $bell(90) > 10^{100}$. This example should illustrate that for an algorithm as the GIW-PHD filter, it is impossible to consider all possible partitions. Therefore, suitable subsets of the partition set $\mathcal{P}$ are necessary for implementations. For an obtained partition set it is necessary that it consists of unique partitions $\wp$. If two partitions are equal e.g. $\wp^{(1)} = \wp^{(2)}$, there is no asset for a filter algorithm, but an additional unnecessary computational effort.

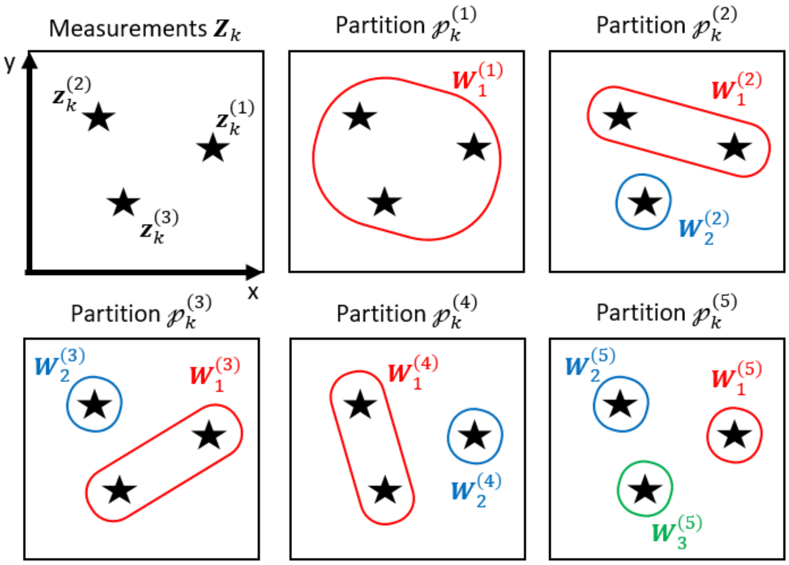

Figure 2: Illustration of the possible partitions with three measurements. The measurements $z_k^{(1)}, z_k^{(2)}$ and $z_k^{(3)}$ are partitioned in five different ways leading to the partitions.

Note that Figure 2 illustrates the example given in Table 4.

6.1.1. Distance partitioning

Distance Partitioning (DP) was supposed in [9], with corrections on the foundational theorem [47][11], to obtain a subset of partitions that contains the most likely partitions. The method relies on the idea that extended objects create spatially adjacent measurements, which can be clustered into a partition set using an upper d_U and lower distance threshold d_L.

For a measurement set $\boldsymbol{Z} = \left\{\boldsymbol{z}^{(i)}\right\}_{i=1}^{n_z}$ a distance measure $d(\cdot,\cdot)$ is used to calculate the distances between every pair of measurements given as

$$\Lambda_{i,j} \triangleq d\left(\boldsymbol{z}^{(i)}, \boldsymbol{z}^{(j)}\right), \text{for } 1 \leq i \neq j \leq n_{\boldsymbol{z}}. \tag{6.1}$$

Then a set of thresholds is determined that starts at the lower threshold d_L and increases the threshold up to d_U stated as

$$\{d_l\}_{l=1}^{n_d}, d_1 = d_L < d_{n_d} = d_U. \tag{6.2}$$

The thresholds can be seen as tuning parameter when no prior information is available about the tracking scenario. A reasonable choice of these parameters is given in the implementation in chapter 8. How to choose the threshold in a theoretically optimal way is discussed in [9] by using a unit-less distance thresholds in combination with the $\mathcal{X}^2$ distribution and the Mahalanobis distance.

The Distance Partitioning algorithm for obtaining a subset of partitions is given in Table 8. For the algorithm an assignment variable $\boldsymbol{\varphi}$ is introduced that facilitates the description and implementation of the partitioning algorithm. $\boldsymbol{\varphi}$ is a vector of length $n_{\boldsymbol{z}}$ with entries φ_m. $\boldsymbol{\varphi}$ indicates which identifier a measurement is associated to:

[11] The original theorem in [9] that proofs the uniqueness of the partitions obtained using distance partitioning was erroneous and therefore a correction is given in [47].

$$\boldsymbol{\varphi} = \left(\varphi_{m_1}, \dots, \varphi_{m_{n_z}}\right). \tag{6.3}$$

In general, an identifier can be the index of an object estimate. In the distance partitioning algorithm, the identifier are simple integers that indicate the cell in which the measurement is assigned. Note that for the extended object PHD correction it does not matter what specific identifiers the single measurements get. An example:
Let the assignment vector $\boldsymbol{\varphi}^{(1)} = (1,1,2,3)$ and $\boldsymbol{\varphi}^{(2)} = (5,5,2,7)$ partition the measurement set $\boldsymbol{Z} = \{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(4)}\}$. Both $\boldsymbol{\varphi}^{(1)}$ and $\boldsymbol{\varphi}^{(2)}$ have equivalent partitioning $\wp\left(\boldsymbol{\varphi}^{(1)}\right) = \wp\left(\boldsymbol{\varphi}^{(2)}\right)$, because both imply the same partition

$$\wp^{(1)} = \left\{\left\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}\right\}, \left\{\boldsymbol{z}^{(3)}\right\}, \left\{\boldsymbol{z}^{(4)}\right\}\right\} = \wp^{(2)}. \tag{6.4}$$

Table 8: The distance partitioning algorithm

```
Given a set of distance thresholds $\{d_l\}_{l=1}^{n_d}$, distances of each measurement pair $\Lambda_{i,j}$
set $\mathcal{P} = \emptyset$
for $l = 1, \dots, n_d$
    $\boldsymbol{\varphi} = \boldsymbol{0}$ # set all $\varphi_m$ to zero
    $cellID = 1$ # set current cell identifier to 1
    for $i = 1, \dots, n_{\boldsymbol{z}}$
        if $\varphi_{m_i} = 0$
            $\varphi_{m_i} = cellID$
            $\boldsymbol{\varphi} = FindNeighbors(i, \boldsymbol{\varphi}, cellID, \Lambda_{i,j})$ # FindNeighbors see Table 9
            $cellID = cellID + 1$
        end if
    end for
    add $\wp(\boldsymbol{\varphi})$ to set $\mathcal{P}$
end for
Output partition set $\mathcal{P} = \{\wp^{(1)}, \dots, \wp^{(n_d)}\}$[12]
```

Table 9: Recursive FindNeighbors-function

```
function $\boldsymbol{\varphi} = findNeigbors(i, \boldsymbol{\varphi}, cellID, \Lambda_{i,j})$
for $j = 1, \dots, n_{\boldsymbol{z}}$
    if $i \neq j$ & $\Delta_{i,j} \leq d_l$ & $\varphi_{m_j} = 0$
        $\varphi_{m_j} = cellID$
        $\boldsymbol{\varphi} = FindNeighbors(j, \boldsymbol{\varphi}, cellID, \Lambda_{i,j})$
    end if
end for
end function
```

[12] $\mathcal{P}$ is not guaranteed to be unique. This depends on the used distance thresholds d_l and the steps between. To process $\mathcal{P}$ in the GIW-PHD filter it should be made unique, to avoid unnecessary computation.

The distance partitioning algorithm (Table 8 and Table 9) was used in [42] and [6]. The results have shown to work well when the objects are spatially separated enough according to the lower distance threshold d_L. Nevertheless, when the objects are spatially too close, they can not be separated and the filter shows cardinality errors, because DP puts close objects into the same cell. To improve this drawback, a further method called Sub-Partitioning is proposed in [9]. This method can be performed after Distance Partitioning and is based on the expected number of measurements per objects.

6.1.2. Sub-partitioning

Sub-Partitioning (SP) can be used to improve the partitions computed with DP. SP computes, if necessary, additional partitions by verifying each cell $\boldsymbol{W}$ in a partition $\mathscr{p}$, if the number of detections in the cell is larger than the expected number of measurements. This can happen, if measurements of more than one object are clustered into one cell.

After a partition set $\mathcal{P} = \left\{\mathscr{p}^{(p)}\right\}_{n=1}^{|\mathcal{P}|}$ is computed using DP in Table 8, SP uses the maximum likelihood estimates $\hat{n}_{\boldsymbol{x}}^{i}$ of the number of objects in the ith cell $\boldsymbol{W}_i^{(p)}$ of every partition $\mathscr{p}^{(p)} \in \mathcal{P}$. If the estimate is larger than one, the cell $\boldsymbol{W}_i^{(p)}$ has to be split into $\hat{n}_{\boldsymbol{x}}^{i}$ smaller cells, denoted as

$$\{\boldsymbol{W}_s^{+}\}_{s=1}^{\hat{n}_x^i}. \tag{6.5}$$

The split cells in (6.5) with the older cells in the partition $\mathscr{p}^{(p)}$ are then added as new partition $\mathscr{p}^{(+)}$ to the partition set $\mathcal{P}$. The Sub-Partitioning algorithm of [9] is given in Table 10, where the function to split cells is denoted as

$$split\left(\hat{n}_{\boldsymbol{x}}^{i}, \boldsymbol{W}_i^{(p)}\right). \tag{6.6}$$

In this thesis, for the $split(\cdot)$ function K-means++ was used, see e.g. chapter 9.1 of textbook [48]. The computation of $\hat{n}_{\boldsymbol{x}}^{i}$ is based on the expected number of measurements per object γ, cf. Subsection 5.3.1 assumption A.9, which is modelled as Poisson RFS. Therefore, the Poisson pdf with the measurement rate γ[13] determines the likelihood function for the number of objects in a cell $W_i^{(p)}$ given as

$$p\left(\left|\boldsymbol{W}_i^{(p)}\right| \middle| n_{\boldsymbol{x}}^{i} = n\right) = \mathcal{PS}\left(\left|\boldsymbol{W}_i^{(p)}\right|, \gamma \cdot n\right). \tag{6.7}$$

$\left|\boldsymbol{W}_i^{(p)}\right|$ denotes the number of measurements in the ith cell of the pth partition, where it is assumed that the volume covered by the cell is sufficiently small. $n_{\boldsymbol{x}}^{i}$ denotes the number of objects in cell $\boldsymbol{W}_i^{(p)}$. Thus the number of clutter measurements in the cell is negligible. The maximum likelihood estimate $\hat{n}_{\boldsymbol{x}}^{i}$ can be calculated as

$$\hat{n}_{\boldsymbol{x}}^{i} = \arg\max_{n} p\left(\left|\boldsymbol{W}_i^{(p)}\right| \middle| n_{\boldsymbol{x}}^{i} = n\right). \tag{6.8}$$

Note that in [9] it is remarked that other alternatives could be found to find the maximum-likelihood estimate for the object number. For the $split(\cdot)$ function in (6.6), there could be other conceivable alternatives, but the simulation results in [9] showed that DP in combination SP using K-means++ delivers improved results compared to the single use of DP.

[13] For unknown measurement rates γ see [9, p. 10], [62]

Table 10: The sub-partitioning algorithm

```
Given unique partition set $\mathcal{P} = \{\wp^{(1)}, \dots, \wp^{(|\mathcal{P}|)}\}$, $|\mathcal{P}|$ is the number of partitions
$l = |\mathcal{P}|$ # initialize counter for new partitions
for $p = 1, \dots, |\mathcal{P}|$
    for $i = 1, \dots, |\wp^{(p)}|$ # counter through every cell $i$ of $p$th partition
        $\hat{n}_{\boldsymbol{x}}^{i} = \arg\max_{n} p\left(\left|\boldsymbol{W}_i^{(p)}\right| \middle| n_{\boldsymbol{x}}^{i} = n\right)$
        if $\hat{n}_{\boldsymbol{x}}^{i} > 1$
            $l = l + 1$ # increase partition counter
            $\wp^{(l)} = \wp^{(j)} \backslash \boldsymbol{W}_i^{(p)}$ # current partition $j$ except the current cell
            $\{\boldsymbol{W}_s^{+}\}_{s=1}^{\hat{n}_{\boldsymbol{x}}^{i}} = \text{split}\left(\hat{n}_{\boldsymbol{x}}^{i}, \boldsymbol{W}_i^{(p)}\right)$ # split the current cell into $\hat{n}_{x}^{i}$ subcells
            $\wp^{(l)} = \wp^{(j)} \cup \{\boldsymbol{W}_s^{+}\}_{s=1}^{\hat{n}_{x}^{i}}$ # add the new cells to the current partition
            add $\wp^{(l)}$ to set $\mathcal{P}$ # add the new partition to the set
        end if
    end for
end for
Output augmented partition set $\mathcal{P}^{+} = \{\wp^{(1)}, \dots, \wp^{(l)}\}$
```

6.1.3. Further cluster approaches

Beside the partitioning methods DP and SP, in [6] and [9] further methods are considered and used in simulations and real world data application. For the sake of completeness, they are named here shortly:

In [9] one alternative method that was investigated is K-means [48] as clustering method. This algorithm needs the number of desired clusters K in advance – hence the name K-means. The problem that arises is, how to choose K in a tracking scenario. One way would be to use a set of different Ks, containing the sequence from one to a certain number of expected objects $\hat{n}_{\boldsymbol{x}}$, denoted as $\{K\}_{K=1}^{\hat{n}_x}$. This was simulated in [9] with the conclusion that the computational load is higher compared to the use of DP and the choice of the set of Ks is less intuitve. In [6] the author further proposed a method called prediction partition. This method uses the predicted filter components to evaluate with the predicted mean of an object in order to obtain where measurements can occur. Then the Mahalanobis distance with a certain threshold is used to cluster the measurements around the predicted mean. Further, a second method called Expectation Maximization (EM) is mentioned, where further details are given in chapter 9 of [48].

Beside these methods, further heuristic methods exist that are more or less common. One common method is DBSCAN [49] (Density-Based Spatial Clustering of Applications with Noise). The advantage of DBSCAN is that the algorithm is capable to identify outliers that are not part of a dense cluster. One drawback is that DBSCAN determines clusters with a fixed parameter, which defines the distance to adjacent points. In scenarios where the density of the points or measurements differs much from object to object, DBSCAN has poor performance. Out of this reason an advanced algorithm is mentioned here called OPTICS [50] (Ordering Points To Identify the Clustering Structure). This can be seen as an advancement to DBSCAN because OPTICS is capable of clustering points or measurements with different spatial densities.

Especially for the GIW-PHD filter some effort was made in the last years to improve the partitioning methods so that they handle closely spaced objects in a satisfying way. In 2014 in [51] the Bayesian theorem was combined with the fuzzy adaptive resonance theory, which is a neural network architecture. In 2015 in [52] the authors combine the DBSCAN clustering with the Fuzzy C-Means method. With this combined cluster method, the risk of faulty clustering through hard decision should be reduced. in 2016 in [53] the problem of closely spaced

manoeuvring objects was investigated with the so called Shape Selection Partitioning. The underlying idea is to combine predicted object centres with different extent shapes and to cluster the measurements according to the possible combinations.
The filter implementations of the mentioned three approaches show, compared to the original work, only slightly better performance but at the cost of higher computational cost.

To conclude the review of classical partitioning methods of section 6.1, the drawbacks and potential improvements are summarized. Some partitioning methods need a clustering parameter. They are dependent on the tracking scenario and have to be adopted manually, which directly influences the filter performance. In scenarios where objects have enough distance between each other even simple clustering algorithms like DP deliver good performance with a low computational cost. But all methods are more or less heuristic approaches that do not consider the multi-object likelihood $p(\boldsymbol{X}|\boldsymbol{Z})$ and have problems with really closely spaced objects. Granström grabbed the idea of incorporating the multi-object likelihood which leads to likelihood-based data association methods that are introduced in the next section.

6.2. Likelihood-based clustering and data association

The likelihood-based data association approach, which is used in this thesis to derive a novel partitioning method for the GIW-PHD filter, was proposed by Granström et al. in 2017 [14] with an extended version in 2018 [10]. The method is called Stochastic Optimisation (SO) and is based on Markov Chain Monte Carlo (MCMC) methods. The proposed method is a 1-step approach, which means the clustering and assignment is done simultaneously, whereas classical methods first cluster and then assign the measurements to objects.

6.2.1. Basic idea of sampling methods

MCMC methods are used to randomly sample values from a distribution, which normally is not known with all its properties. This distribution is frequently the posterior density in Bayesian interference [54]. MCMC methods construct a Markov chain whose stationary distribution is equal to the probability distribution of interest, here $P\{\mathcal{A}\}$, where $\mathcal{A}$ corresponds to the data association. In context of object tracking, it is not of interest generating samples from the multi-object distribution but finding highly probable data associations $\mathcal{A}$ for existing measurements at every time step that maximise this distribution [14].
The general idea of the sampling method is as follows: Start with an initial data association for the measurements. Then perform a random action out of defined possible actions that change the initial association. For every performed action, the multi-object likelihood can be evaluated by exploiting the Bayes update. The higher the resulting likelihood of a certain action, the higher the probability of selecting the association that arises from that certain action. Finally, the selected association is the initial association for the next iteration. By repeating the described procedure for several iterations, the sampling approach finds highly probable associations.
The sampling method SO proposed by Granström et al. was inspired by previous works [55] [56], where the stochastic sampling methods Gibbs sampling and Metropolis Hastings (MH) were used. SO combines the actions of Gibbs sampling and MH. Gibbs sampling only effects one measurement whereas MH affects all measurements in a cell, therefore the idea behind the combination of both sampling methods in SO is to obtain a faster algorithm that requires fewer iterations [10].
The benefits of using stochastic sampling methods in contrast to classical 2-step method is that it works directly on the multi-object likelihood, which is the quantity that should be maximised with the choice of a probable data association. The method further works well for distant and

close objects and finally there is only one parameter that has to be set – the number of iterations, which the sampling algorithm runs per time step k.

6.2.2. Stochastic Optimisation

The SO sampling method proposed in [10] is applied to a Poisson Multi-Bernoulli Mixture (PMBM) filter [57] [58]. For the sake of clarity, the presentation of the SO sampling algorithm in this subsection is simplified. Equations and declarations that are PMBM filter specific, are dropped, i.e. are kept as general as possible to adapt them later to the GIW-PHD filter.

Let the measurements collected at time step k be the set $\boldsymbol{Z}_k$ and the single measurements be indexed by $m \in \mathbb{M}_k$, where $\mathbb{M}_k$ is the index set of the measurements. Thus, the measurement set is $\boldsymbol{Z}_k = \left\{\boldsymbol{z}_k^{(m)}\right\}_{m \in \mathbb{M}_k}$. Further let $\mathcal{A}$ be an association that assigns each measurement in $\boldsymbol{Z}_k$ to a source, which either is the background or one existing object. Note that in the context of SO with a PMBM filter, the background source could be clutter or a new object. Further, in the PMBM filter each object modelled as Bernoulli RFS has a specific index i. For the GIW-PHD filter in its roots there is no specific object labeling, but for a general explanation it is kept, so that the SO method can be applied to every filter with specific object data association as well. Thus, the existing objects are labelled with $i \in \mathbb{I}_{k|k-1}$, where $\mathbb{I}_{k|k-1}$ is the index set of predicted objects before the filter update step. For further explanations, the time index $k|k-1$ is dropped since all iterations of SO take place at a certain time step k.

Formally, a data association $\mathcal{A}$ consists of a partition of the united set $\mathbb{M} \cup \mathbb{I}$ into non-empty disjoint subcells and it holds $\mathbb{M} \cap \mathbb{I} = \emptyset$.

An example: There are three objects and four measurements, i.e. $\mathbb{I} = \{i_1, i_2, i_3\}$ and $\mathbb{M} = \{m_1, m_2, m_3, m_4\}$. One valid association $\mathcal{A}$ or partition $\mathbb{M} \cup \mathbb{I}$ is

$$\mathcal{A}: \{i_1, m_1\}, \{i_2, m_2, m_3\}, \{i_3\}, \{m_4\}. \tag{6.9}$$

Here the measurement m_1 is assigned to object i_1, measurements m_2 and m_3 are assigned to object i_2, object i_3 has no assigned measurement, i.e. is not detected and finally measurement m_4 is clutter or a new object. Again, the assignment variable $\boldsymbol{\varphi}$ is used, which is a vector with length $|\boldsymbol{Z}|$ and single entries φ_m. If $\varphi_m \in \mathbb{I}$, then the measurement with index m, i.e. $\boldsymbol{z}^{(m)}$, is associated to the object with index φ_m and if $\varphi_m \notin \mathbb{I}$[14] then the measurement $\boldsymbol{z}^{(m)}$ is associated to the background. Two measurements $\boldsymbol{z}^{(m_1)}$ and $\boldsymbol{z}^{(m_2)}$ are associated to the same object if $\varphi_{m_1} = \varphi_{m_2} \in \mathbb{I}$. A measurement cell with the same source is denoted as

$$\boldsymbol{C}_c = \bigcup_{m:\varphi_m = c} \boldsymbol{z}^{(m)}, \tag{6.10}$$

where c is a specific entry of φ_m. For a given assignment vector $\boldsymbol{\varphi}$, the equivalent association $\mathcal{A}(\boldsymbol{\varphi})$ is obtained by forming subsets of measurements that have equal assignments with indices $m: \varphi_m = c$ and including the object index if $\varphi_m \in \mathbb{I}$, see (6.9).

The SO sampling algorithm runs over a certain number of iterations. Starting with an arbitrary initial assignment $\boldsymbol{\varphi}^{(t)}$. Let t be the number of the current iteration, the next assignment $\boldsymbol{\varphi}^{(t+1)}$ is obtained by performing the SO procedure:

First, randomly sample a measurement index r_1. The corresponding cell is denoted as $\boldsymbol{C}_{c_1}^{(t)}$ with $\varphi_{r_1}^{(t)} = c_1$. Second, an action α is performed to the cell, obtaining a new different assignment vector $\boldsymbol{\varphi}^{(t+1)}$, i.e. $\boldsymbol{\varphi}^{(t+1)} = \boldsymbol{\varphi}_\alpha^{(t)}$. The corresponding randomly sampled measurement association is $\mathcal{A}_\alpha^{(t)} = \mathcal{A}\left(\boldsymbol{\varphi}_\alpha^{(t)}\right)$. The actions that can be performed are given in Table 11 with examples,

[14] Entries for $\varphi_m \notin \mathbb{I}$ are denoted with $c', c'', \dots$

where the abbreviation $\mathbb{S}^{(t)} = \mathbb{I} \cup \left\{\varphi_m^{(t)}\right\}$ is used according to [14], which implies $\mathbb{S}^{(t)}$ to be an index set of unique entries that consists of the single object indices combined with the single vector element $\left\{\varphi_m^{(t)}\right\}$.

Table 11: Stochastic Optimisation actions

Given: Initial assignment vector $\boldsymbol{\varphi}^{(t)}$[15], randomly sampled measurement index r_1 that determines $c_1 = \varphi_{m=r_1}$

Action 1 - Do nothing

$$\boldsymbol{\varphi}^{(t+1)} = \boldsymbol{\varphi}^{(t)} \tag{6.11}$$

Action 2 – Move one measurement into an existing cell
If $\left|\boldsymbol{C}_{c_1}^{(t)}\right| > 1$, move $\boldsymbol{z}^{(r)}$ to cell $\boldsymbol{C}_{c_2}^{(t)}$, where $c_2 \in \mathbb{S}^{(t)} \backslash c_1$:

$$\varphi_m^{(t+1)} = \begin{cases} c_2, & \text{if } m = r \\ \varphi_m^{(t)}, & \text{otherwise} \end{cases} \tag{6.12}$$

Possible actions[16]*:* There are $\left|\mathbb{S}^{(t)} \backslash c_1\right| = \left|\mathbb{S}^{(t)}\right| - 1$ other cells to which the measurement can be moved.
Example for one possible action: Given objects $\{i_1, i_2\} \in \mathbb{I}$, measurements $\left\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \boldsymbol{z}^{(3)}, \boldsymbol{z}^{(4)}\right\}$, $\boldsymbol{\varphi}^{(t)} = (i_1, i_1, i_2, c')$. Sampled $r_1 = 2$ evaluates $\varphi_{m_2}^{(t)} = c_1 = i_1$. Select e.g. $c_2 = \varphi_{m_4}^{(t)} = c'$ and change as in (6.12) $\boldsymbol{\varphi}^{(t+1)} = [i_1, c', i_2, c']$ with corresponding cells $\boldsymbol{C}_{c_1}^{(t)} = \left\{\boldsymbol{z}^{(1)}\right\}$, $\boldsymbol{C}_{c_2}^{(t)} = \left\{\boldsymbol{z}^{(3)}\right\}$, $\boldsymbol{C}_{c'}^{(t)} = \left\{\boldsymbol{z}^{(2)}, \boldsymbol{z}^{(4)}\right\}$. Resulting association $\mathcal{A}_\alpha^{(t)}: \{i_1, m_1\}, \{i_2, m_3\}, \{m_2, m_4\}$.

Action 3 - Move one single measurement to a new cell
If $\left|\boldsymbol{C}_{c_1}^{(t)}\right| > 1$, move $\boldsymbol{z}^{(r)}$ to new cell $\boldsymbol{C}_{c_\star}^{(t+1)}$, where $c_\star \notin \mathbb{S}^{(t)}$:

$$\varphi_m^{(t+1)} = \begin{cases} c_\star, & \text{if } m = r \\ \varphi_m^{(t)}, & \text{otherwise} \end{cases} \tag{6.13}$$

Possible actions: Moving the measurement into a new cell is one possible action.
Example: Given objects $\{i_1, i_2\} \in \mathbb{I}$, measurements $\left\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \boldsymbol{z}^{(3)}, \boldsymbol{z}^{(4)}\right\}$, $\boldsymbol{\varphi}^{(t)} = (i_1, i_1, i_2, c')$. Sampled $r_1 = 1$ evaluates $\varphi_{m_1}^{(t)} = c_1 = i_1$. Check if $\left|\boldsymbol{C}_{c_1}^{(t)}\right| > 1$, then change $\varphi_{m_1}^{(t)} = c_\star$ according to (6.13). Changed $\boldsymbol{\varphi}^{(t+1)} = (c_\star, i_1, i_2, c')$ with corresponding measurement cells $\boldsymbol{C}_{c_1}^{(t)} = \left\{\boldsymbol{z}^{(2)}\right\}$, $\boldsymbol{C}_{c_2}^{(t)} = \left\{\boldsymbol{z}^{(3)}\right\}$, $\boldsymbol{C}_{c_\star}^{(t+1)} = \left\{\boldsymbol{z}^{(1)}\right\}$. Resulting association $\mathcal{A}_\alpha^{(t)}: \{i_1, m_2\}, \{i_2, m_3\}, \{m_1\}, \{m_4\}$.

Action 4 – Merge two cells into one cell
Sample additional measurement index r_2 then merge cell $\boldsymbol{C}_{c_1}^{(t)}$ and $\boldsymbol{C}_{c_2}^{(t)}$, where $c_2 \in \mathbb{S}^{(t)} \backslash c_1$:

$$\varphi_m^{(t+1)} = \begin{cases} c_2, & \text{if } \varphi_m^{(t)} = c_1 \\ \varphi_m^{(t)}, & \text{otherwise} \end{cases} \tag{6.14}$$

Possible actions: There are $\left|\mathbb{S}^{(t)} \backslash c_1\right| = \left|\mathbb{S}^{(t)}\right| - 1$ other cells with which it is possible to merge.
Example for one possible action: Given objects $\{i_1, i_2, i_3\} \in \mathbb{I}$, measurements $\left\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \boldsymbol{z}^{(3)}, \boldsymbol{z}^{(4)}, \boldsymbol{z}^{(5)}\right\}$, $\boldsymbol{\varphi}^{(t)} = (i_3, i_1, i_1, c', i_2)$. Sampled $r_1 = 4, r_2 = 3$ evaluates

[15] SO is not sensitive to initialization – the choice of $\boldsymbol{\varphi}^{(t)}$ can be arbitrary or according to [14, p. 40].

[16] Possible actions means all $\left|\mathbb{S}^{(t)}\right| - 1$ varieties of action one have to be considered, which leads to $\left|\mathbb{S}^{(t)}\right| - 1$ different vectors $\boldsymbol{\varphi}^{(t+1)}$.

$\varphi_{m_4}^{(t)} = c_1 = c', \varphi_{m_3}^{(t)} = c_2 = i_1$. Change $\varphi_{m_4}^{(t)} = i_1$ according to (6.14). Changed $\boldsymbol{\varphi}^{(t+1)} = (i_3, i_1, i_1, i_1, i_2)$ with corresponding cells $\boldsymbol{C}_{c_1}^{(t)} = \emptyset, \boldsymbol{C}_{c_2}^{(t)} = \{\boldsymbol{z}^{(2)}, \boldsymbol{z}^{(3)}, \boldsymbol{z}^{(4)}\}$. Resulting association $\mathcal{A}_\alpha^{(t)}: \{i_i, m_2, m_3, m_4\}, \{i_2, m_5\}, \{i_3, m_1\}$.

Action 5 –Move all measurements of a cell with assigned object into new cell
If $c_1 \in \mathbb{I}$, change assignments to $c_\star \notin \mathbb{S}^{(t)}$:

$$\varphi_m^{(t+1)} = \begin{cases} c_\star, & \varphi_{\boldsymbol{m}}^{(t)} = c_1 \\ \varphi_m^{(t)}, & \text{otherwise} \end{cases} \tag{6.15}$$

Possible actions: Moving the measurement into a new cell is one possible action.
Example: Given objects $\{i_1, i_2\} \in \mathbb{I}$, measurements $\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \boldsymbol{z}^{(3)}, \boldsymbol{z}^{(4)}, \boldsymbol{z}^{(5)}\}$, $\boldsymbol{\varphi}^{(t)} = (c'', i_1, i_1, c', i_2)$. Sampled $r_1 = 3$ evaluates $\varphi_{m_3}^{(t)} = c_1 = i_1 \in \mathbb{I}$. Change $\varphi_{m_2} = c_\star$ and $\varphi_{m_3} = c_\star$ according to (6.15). Changed $\boldsymbol{\varphi}^{(t+1)} = (c'', c_\star, c_\star, c', i_2)$ with resulting association $\mathcal{A}_\alpha^{(t)}: \{i_1\}, \{i_2, m_5\}, \{m_1\}, \{m_2, m_3\}, \{m_4\}$.

Action 6 – Split cell into two sub-cells and assign one sub-cell to new cell
If $\left|\boldsymbol{C}_{c_1}^{(t)}\right| > 1$, first split $\boldsymbol{C}_{c_1}^{(t)}$ into $\boldsymbol{C}_{c_1,1}^{(t)}$ and $\boldsymbol{C}_{c_1,2}^{(t)}$, and proceed with $c_\star \notin \mathbb{S}^{(t)}$
a) if $c_1 \notin \mathbb{I}$, then

$$\varphi_m^{(t+1)} = \begin{cases} c_\star, & \text{if } \boldsymbol{z}_k^{(m)} \in \boldsymbol{C}_{c_1,2}^{(t)} \\ \varphi_m^{(t)}, & \text{otherwise} \end{cases} \tag{6.16}$$

b) else if $c_1 \in \mathbb{I}$, then select one of the two alternatives

$$\varphi_m^{(t+1)} = \begin{cases} c_\star, & \text{if } \boldsymbol{z}_k^{(m)} \in \boldsymbol{C}_{c_1,1}^{(t)} \\ \varphi_m^{(t)}, & \text{otherwise} \end{cases} \tag{6.17}$$

$$\varphi_m^{(t+1)} = \begin{cases} c_\star, & \text{if } \boldsymbol{z}_k^{(m)} \in \boldsymbol{C}_{c_1,2}^{(t)} \\ \varphi_m^{(t)}, & \text{otherwise,} \end{cases} \tag{6.18}$$

Possible actions: For a cell split[17] there is either one possible action, cf. (6.16) or two possible actions, cf. (6.17) and (6.18).
Examples:
Given objects $\{i_1, i_2\} \in \mathbb{I}$, measurements $\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \boldsymbol{z}^{(3)}, \boldsymbol{z}^{(4)}, \boldsymbol{z}^{(5)}\}$, $\boldsymbol{\varphi}^{(t)} = (c', c', i_1, i_2, i_1)$.
a) Sampled $r_1 = 2$ evaluates $\varphi_{m_2}^{(t)} = c_1 = c' \notin \mathbb{I}$. Split $\boldsymbol{C}_{c_1}^{(t)} = \{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}\}$ results in subcells $\boldsymbol{C}_{c_1,1}^{(t)} = \{\boldsymbol{z}^{(1)}\}$ and $\boldsymbol{C}_{c_1,2}^{(t)} = \{\boldsymbol{z}^{(2)}\}$. Change $\varphi_{m_2} = c_\star$ according to(6.16). Resulting $\boldsymbol{\varphi}^{(t+1)} = [c', c_\star, i_1, i_2, i_1]$ with corresponding association $\mathcal{A}_\alpha^{(t)}: \{i_1, m_3, m_5\}, \{i_2, m_4\}, \{m_1\}, \{m_2\}$.
b) Sampled $r_1 = 5$ evaluates $\varphi_{m_5}^{(t)} = c_1 = i_1 \in \mathbb{I}$. Split $\boldsymbol{C}_{c_1}^{(t)} = \{\boldsymbol{z}^{(3)}, \boldsymbol{z}^{(5)}\}$ results in subcells $\boldsymbol{C}_{c_1,1}^{(t)} = \{\boldsymbol{z}^{(3)}\}$ and $\boldsymbol{C}_{c_1,2}^{(t)} = \{\boldsymbol{z}^{(5)}\}$.
Alternative 1: Change $\varphi_{m_3} = c_\star$ according to (6.17). Resulting $\boldsymbol{\varphi}^{(t+1)} = (c', c', c_\star, i_2, i_1)$ with corresponding association $\mathcal{A}_\alpha^{(t)}: \{i_1, m_5\}, \{i_2, m_4\}, \{m_1, m_2\}, \{m_3\}$.
Alternative 2: Change $\varphi_{m_5} = c_\star$ according to(6.18). Resulting $\boldsymbol{\varphi}^{(t+1)} = (c', c', i_1, i_2, c_\star)$ with corresponding association $\mathcal{A}_\alpha^{(t)}: \{i_1, m_3\}, \{i_2, m_4\}, \{m_1, m_2\}, \{m_5\}$.

[17] E.g. using K-means++ [48].

Third, after the applied SO actions of Table 11, n_α different assignment vectors $\boldsymbol{\varphi}_\alpha^{(t)}$with corresponding associations $\mathcal{A}_\alpha^{(t)}$ are obtained. The total number of different assignments that results out of the actions in Table 11 is given as [14]

$$n_\alpha = \begin{cases} |\mathbb{S}^t| + 1, & \text{if } \left|\boldsymbol{C}_{c_1}^{(t)}\right| = 1 \\ 2|\mathbb{S}^t| + 2, & \text{if } \left|\boldsymbol{C}_{c_1}^{(t)}\right| > 1 \notin \mathbb{I} \\ 2|\mathbb{S}^t| + 3, & \text{if } \left|\boldsymbol{C}_{c_1}^{(t)}\right| > 1 \in \mathbb{I}. \end{cases} \tag{6.19}$$

The different associations have to be evaluated due to their impact on the multi-object likelihood, which leads to a relative acceptance likelihood $\mathcal{L}^{\mathcal{A}_\alpha^{(t)}}$ of each association, given as

$$P\left(\boldsymbol{\varphi}^{(t+1)} = \boldsymbol{\varphi}_\alpha^{(t)} \middle| \boldsymbol{\varphi}^{(t)}\right) = \frac{\mathcal{L}^{\mathcal{A}_\alpha^{(t)}}}{\sum_{\alpha'} \mathcal{L}^{\mathcal{A}_{\alpha'}^{(t)}}}. \tag{6.20}$$

Thus, with probability $\mathcal{L}^{\mathcal{A}_\alpha^{(t)}}$ an association is sampled for the next iteration. The sampled association determines the assignment vector $\boldsymbol{\varphi}^{(t+1)} = \boldsymbol{\varphi}_\alpha^{(t)}$ for the next iteration. The calculation of $\mathcal{L}^{\mathcal{A}_\alpha^{(t)}}$ is filter dependent and later shown in case of the GIW-PHD filter.

The SO procedure explained in these three steps samples data associations with high posterior likelihood more often. A subset of data associations that are used for a filter update set, is obtained by scoring the associations in order of their likelihood and choosing the most likely ones. In the case of the GIW-PHD filter the relative likelihood of a data association corresponds to the relative weight of a single partition, which is depicted in the next subsection.

6.2.3. Stochastic Partitioning

In the original work [10], the SO method presented in subsection 6.2.2 is applied to a PMBM filter and can not directly be applied to the GIW-PHD filter framework. This section presents a likelihood-based partitioning method that is based on SO. The proposed method is called Stochastic Partitioning (StP) – partitioning, because the GIW-PHD filter framework processes a subset of partitions in its update step, cf. section 5.3, and within StP the partitions are changed doing stochastic steps. The integration into the GIW-PHD filter is presented in chapter 7.

The procedure of Stochastic Partitioning is related to SO but with slightly different single actions α. Let the measurement set of filter step k be $\boldsymbol{Z}_k = \left\{\boldsymbol{z}_k^{(m)}\right\}_{m \in \mathbb{M}_k}$, where a partition $\wp^{(t)}$ is built using the assignment vector as $\wp^{(t)} = \wp(\boldsymbol{\varphi}^{(t)})$. The single cells are now called $\boldsymbol{W}$ to obtain the notation in context of partitions. Therefore, a measurement cell is, analogous to (6.10), defined as

$$\boldsymbol{W}_c^{(\wp)} = \bigcup_{m:\varphi_m = c} \boldsymbol{z}^{(m)}, \tag{6.21}$$

where the superscript denotes the partition and the subscript the cell identifier. Note that for the GIW-PHD filter no explicit object assignment is used so that the index set $\mathbb{I}$ of SO is dropped. That means for the application of StP, that $\mathbb{S}^{(t)} = \bigcup_m \left\{\varphi_m^{(t)}\right\}$.

The StP algorithm runs over a number of iterations T, starting with an initial assignment, expressed by the assignment vector $\boldsymbol{\varphi}^{(t)}$ with corresponding measurement set partition $\wp^{(t)} = \wp(\boldsymbol{\varphi}^{(t)})$. In each iteration actions are applied to the current assignment $\boldsymbol{\varphi}^{(t)}$ leading to several new assignment vectors $\boldsymbol{\varphi}_\alpha^{(t)}$ with corresponding partitions $\wp_\alpha^{(t)} = \wp\left(\boldsymbol{\varphi}_\alpha^{(t)}\right)$. The actions differ from those presented in Table 11, because there is no index set $\mathbb{I}$ to be considered, i.e. this affects the number of possible actions as well as the kind of the single actions. The actions of

StP are given in Table 12, where the index $\wp$ of $\boldsymbol{W}_c^{(\wp)}$ is replaced by t because the affected cells by the StP actions are changed over the iterations. The entries c denote the cell membership and can be chosen as plain integers or any other symbols. Equal entries $\varphi_m = c$ in the assignment vector denote the same measurement cell. The abbreviation $\mathbb{S}^{(t)} = \bigcup_m \left\{\varphi_m^{(t)}\right\}$ in Table 12 implies $\mathbb{S}^{(t)}$ to be a set with the unique entries of the assignment vector $\boldsymbol{\varphi}^{(t)}$, which is used to determine the number of possible actions.

Table 12: Stochastic Partitioning actions

Given: Initial assignment vector $\boldsymbol{\varphi}^{(t)}$**,** randomly sampled measurement index r_1 that determines $c_1 = \varphi_{m=r_1}$

Action 1 - Do nothing

$$\boldsymbol{\varphi}^{(t+1)} = \boldsymbol{\varphi}^{(t)} \tag{6.22}$$

Action 2 – Move one measurement into an existing cell

If $\left|\boldsymbol{W}_{c_1}^{(t)}\right| > 1$, move $\boldsymbol{z}^{(r)}$ to cell $\boldsymbol{W}_{c_2}^{(t)}$, where $c_2 \in \mathbb{S}^{(t)} \backslash c_1$:

$$\varphi_m^{(t+1)} = \begin{cases} c_2, & \text{if } m = r \\ \varphi_m^{(t)}, & \text{otherwise} \end{cases} \tag{6.23}$$

Possible actions: There are $\left|\mathbb{S}^{(t)} \backslash c_1\right| = \left|\mathbb{S}^{(t)}\right| - 1$ other cells to which the measurement can be moved.

Example for one possible action: Given measurements $\left\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \boldsymbol{z}^{(3)}, \boldsymbol{z}^{(4)}\right\}$, $\boldsymbol{\varphi}^{(t)} = (c', c'', c''', c''')$ Sampled $r_1 = 3$ evaluates $\varphi_{m_3}^{(t)} = c_1 = c'''$. Select e.g. $c_2 = \varphi_{m_1}^{(t)} = c'$ and change according to (6.23) $\boldsymbol{\varphi}^{(t+1)} = (c', c'', c', c''')$ with corresponding partition $\wp_\alpha^{(t)}: \{m_1, m_3\}, \{m_2\}, \{m_4\}$.

Action 3 - Move one single measurement to a new cell

If $\left|\boldsymbol{W}_{c_1}^{(t)}\right| > 1$, move $\boldsymbol{z}^{(r)}$ to new cell $\boldsymbol{W}_{c_\star}^{(t+1)}$, where $c_\star \notin \mathbb{S}^{(t)}$:

$$\varphi_m^{(t+1)} = \begin{cases} c_\star, & \text{if } m = r \\ \varphi_m^{(t)}, & \text{otherwise} \end{cases} \tag{6.24}$$

Possible actions: Moving one measurement into a new cell is one possible action.

Example: Given measurements $\left\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \boldsymbol{z}^{(3)}, \boldsymbol{z}^{(4)}\right\}$, $\boldsymbol{\varphi}^{(t)} = (c', c'', c''', c''')$. Sampled $r_1 = 3$ evaluates $\varphi_{m_3}^{(t)} = c_1 = c'''$. Change $\varphi_{m_3}^{(t)} = c_\star$ according to (6.24) results in $\boldsymbol{\varphi}^{(t+1)} = (c', c'', c_\star, c''')$ with corresponding partition $\wp_\alpha^{(t)}: \{m_1\}, \{m_2\}, \{m_3\}, \{m_4\}$.

Action 4 – Merge two cells into one cell

Sample additional measurement index r_2, then merge cell $\boldsymbol{W}_{c_1}^{(t)}$ and $\boldsymbol{W}_{c_2}^{(t)}$, where $c_2 \in \mathbb{S}^{(t)} \backslash c_1$:

$$\varphi_m^{(t+1)} = \begin{cases} c_2, & \text{if } \varphi_m^{(t)} = c_1 \\ \varphi_m^{(t)}, & \text{otherwise} \end{cases} \tag{6.25}$$

Possible actions: There are $\left|\mathbb{S}^{(t)} \backslash c_1\right| = \left|\mathbb{S}^{(t)}\right| - 1$ other cells with which it is possible to merge.

Example for one possible action: Given measurements $\left\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \boldsymbol{z}^{(3)}, \boldsymbol{z}^{(4)}, \boldsymbol{z}^{(5)}\right\}$, $\boldsymbol{\varphi}^{(t)} = (c', c''', c''', c', c'')$. Sampled $r_1 = 1, r_2 = 2$ evaluates $\varphi_{m_1}^{(t)} = c_1 = c'$, $\varphi_{m_2}^{(t)} = c_2 = c'''$. Change all $\varphi_m^{(t)} = c_1$ according to (6.25). Changed $\boldsymbol{\varphi}^{(t+1)} = (c''', c''', c''', c''', c'')$ with corresponding cells partition $\wp_\alpha^{(t)}: \{m_1, m_2, m_3, m_4\}, \{m_5\}$.

Action 5 – Split cell into two sub-cells and assign one sub-cell to new cell
If $\left|\boldsymbol{W}_{c_1}^{(t)}\right| > 1$, first split $\boldsymbol{W}_{c_1}^{(t)}$ into $\boldsymbol{W}_{c_1,1}^{(t)}$ and $\boldsymbol{W}_{c_1,2}^{(t)}$, and proceed with $c_\star \notin \mathbb{S}^{(t)}$

$$\varphi_m^{(t+1)} = \begin{cases} c_\star, & \text{if } \boldsymbol{z}_k^m \in \boldsymbol{W}_{c_1,2}^{(t)}{}^{18} \\ \varphi_m^{(t)}, & \text{otherwise} \end{cases} \tag{6.26}$$

Possible actions: A cell split with a new assigned sub-cell corresponds to one possible action.
Example:
Given measurements $\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \boldsymbol{z}^{(3)}, \boldsymbol{z}^{(4)}, \boldsymbol{z}^{(5)}\}$, $\boldsymbol{\varphi}^{(t)} = (c', c', c'', c'', c'')$.
Sampled $r_1 = 5$ evaluates $\varphi_{m_5}^{(t)} = c_1 = c''$. Split $\boldsymbol{W}_{c_1}^{(t)} = \{\boldsymbol{z}^{(3)}, \boldsymbol{z}^{(4)}, \boldsymbol{z}^{(5)}\}$ results in subcells $\boldsymbol{W}_{c_1,1}^{(t)} = \{\boldsymbol{z}^{(4)}\}$ and $\boldsymbol{W}_{c_1,2}^{(t)} = \{\boldsymbol{z}^{(3)}, \boldsymbol{z}^{(5)}\}$. Change $\varphi_{m_3}, \varphi_{m_5} = c_\star$ with resulting $\boldsymbol{\varphi}^{(t+1)} = (c', c', c_\star, c'', c_\star)$ with corresponding partition $\wp_\alpha^{(t)}$: $\{m_1, m_2\}, \{m_3, m_5\}, \{m_4\}$.

In comparison to the actions of SO in Table 11, the number of actions decreases from six to five different actions in Table 12. This is because (6.15) is not applicable in the context of the GIW-PHD filter, because there are no explicit object assignments $i \in \mathbb{I}$. The total number n_α of different partitions $\wp_\alpha^{(t)}$, which are obtained out of the StP actions in Table 12 per iteration t, is given as

$$n_\alpha = \begin{cases} \left|\mathbb{S}^{(t)}\right|, & \text{if } \left|\boldsymbol{W}_{c_1}^{(t)}\right| = 1 \\ 2\left|\mathbb{S}^{(t)}\right| + 1, & \text{if } \left|\boldsymbol{W}_{c_1}^{(t)}\right| > 1. \end{cases} \tag{6.27}$$

In each iteration after applying the StP actions, a partition is sampled with an acceptance likelihood that is given as the relative likelihood $\mathcal{L}^{\wp_\alpha^{(t)}}$, i.e.

$$P\left(\boldsymbol{\varphi}^{(t+1)} = \boldsymbol{\varphi}_\alpha^{(t)} \middle| \boldsymbol{\varphi}^{(t)}\right) = \mathcal{L}^{\wp_\alpha^{(t)}} \tag{6.28}$$

Using the relative weight $\omega_\wp$ of a partition given in (5.10) the absolute partition weight can be defined without the denominator normalization as

$$\mathcal{W}_\wp = \prod_{\boldsymbol{W} \in \wp} d_{\boldsymbol{W}} = d_{\boldsymbol{W}_1^{(\wp)}} \times d_{\boldsymbol{W}_2^{(\wp)}} \times \ldots \times d_{\boldsymbol{W}_{|\wp|}^{(\wp)}}, \tag{6.29}$$

where $|\wp|$ is the number of cells and cell weight $d_{\boldsymbol{W}}$ is defined in Table 6. Using (6.29), the relative partition weight $\omega_\wp$ can be rewritten and is called relative likelihood $\mathcal{L}^{\wp_\alpha^{(t)}}$ given as

$$\mathcal{L}^{\wp_\alpha^{(t)}} = \frac{\mathcal{W}_\wp}{\sum_{\wp'} \mathcal{W}_{\wp'}}. \tag{6.30}$$

Note that the product of all cell weights in a partition (6.29) as the unnormalised partition weight gives an absolute measure that is used to score the partitions that are obtained over all StP iterations. Further information is given in chapter 7.
The integration of StP into the GIW-PHD filter framework of [6] is not straightforward and therefore outlined in the subsequent chapter. There, the classical filter framework, cf. Table 7, is adapted to the needs of the StP integration and a direct applicable implementation of the StP algorithm is given. In chapter 8 the StP method is tested via Monte Carlo simulations and is compared to the classical partitioning methods DP and SP.

[18] It does not matter, which sub-cell $\boldsymbol{W}_{c_1,1}^{(t)}$ or $\boldsymbol{W}_{c_1,2}^{(t)}$ is assigned to $c_\star$, the resulting partitions would be equal.

7. Stochastic Partitioning GIW-PHD filter implementation

In this chapter, an implementation of the likelihood-based partitioning method Stochastic Partitioning (StP), presented in chapter 6, is developed and integrated into the GIW-PHD filter framework [6]. Therefore, the classical GIW-PHD filter procedure has to be adapted, because with the use of StP it is not possible to separate the filter's partitioning step and correction step distinctively. In section 7.1, the adapted framework is given with the presentation of the slightly changed filtering procedure. Section 7.2 presents the StP algorithm that combines the partitioning and correction in one step using the StP actions presented in chapter 6. Then, it is discussed how to calculate the acceptance likelihood of a partition that is drawn due to a different action. Additionally, it is shown that the partition weight computation can be simplified using the initial partition and changing only several cell weights. The section concludes with the suggestion of the "partial correction" of GIW mixture components that are dependent on certain cells, which again lowers the computational cost. Finally in section 7.3, two comments are given for the StP method. One drawback is mentioned that reveals when the filter prediction performs poorly. Further, the implementation issue computing the partition, cell and component weights is indicated.

7.1. Filter framework

The classical filter framework of the GIW-PHD filter [6] is given in Table 7. For a given measurement set $\boldsymbol{Z}_k = \left\{\boldsymbol{z}_k^{(j)}\right\}_{j=1}^{n_{z_k}}$ at time step k and the posterior PHD $D_{k-1|k-1}(\xi_k)$ the first filter steps are the computation of a partition set $\mathcal{P}$ and the prediction of the GIW mixture components. The partition set $\mathcal{P}$ consists of a subset of partitions $\left\{\wp^{(m)}\right\}_{m=1}^{p}$, which is obtained in [6] with the heuristic partitioning methods DP and SP, see chapter 6. In the next filter step, the correction of the GIW mixture components is done using every single partition $\wp^{(m)}$, which was computed in advance of this step. The number of components that are obtained after the correction is given in (5.29). It can be seen, that the main computational effort of the filter is done in the correction step, where every GIW mixture component is updated with each cell $\boldsymbol{W}$ of all partitions, see [43, p. 3] as well. The pruning and merging of GIW mixture components and the state extraction remain unchanged.

For the integration of StP into the GIW-PHD filter, the framework has to be changed because the partitioning and correction step are joint processes in StP. The filter steps using StP are given in Table 13.

Table 13: Stochastic Partitioning GIW-PHD filter steps

```
Given Sequence of measurement sets $\{\boldsymbol{Z}_k\}_{k=1}^{N}$
Initialize $J_{0|0} = 0$
1:  for $k = 1, \dots, N$
2:      Predict the GIW mixture components of birth and existing objects
3:      Stochastic Partitioning step (see section 7.2)
4:      Prune and merge the corrected components
5:      Extract the estimated objects
6:  end for
Output sequence of estimated object sets $\{\widehat{\boldsymbol{X}}_k\}_{k=1}^{N}$ with GIW mixture components
```

In comparison to the original framework, the prediction step is shifted to the first position followed by the Stochastic Partitioning step, which in a way combines partitioning and correction step. The implementation of the StP step is provided in the next section.

7.2. Stochastic Partitioning algorithm

The procedure of the StP algorithm is given in Table 14, where $\mathcal{C}_{(\cdot)}$ denotes all essential filter components for the correction step, cf. Table 6. The procedure of the StP algorithm starts with the correction using the initial partition $\mathscr{p}^{(t)} = \mathscr{p}\big(\boldsymbol{\varphi}^{(t)}\big)$, thus the corrected components $\mathcal{C}_{k|k}$ are obtained. Followed up, the main loop starts with iteration t applying the StP actions to the initial assignment $\boldsymbol{\varphi}^{(t)}$. The resulting assignments $\left\{\boldsymbol{\varphi}_\alpha^{(t)}\right\}_{\alpha=1}^{n_\alpha}$ with corresponding partitions, then are used to process a partial correction step. The partial correction step (line 9) involves the initial corrected components $\mathcal{C}_{k|k}$ and requires not the whole partition $\mathscr{p}_\alpha^{(t)}$ but merely the affected cells $\boldsymbol{W}_{(\cdot)}^{\mathscr{p}_\alpha^{(t)}}$. This method avoids unnecessary computational effort and is discussed later in this section. Followed by the partial correction, the partition weights $\left\{\mathcal{W}_{\mathscr{p}}^{(\alpha)}\right\}_{\alpha=1}^{n_\alpha}$ are computed as well as the relative likelihood $\mathcal{L}^{\mathscr{p}_\alpha^{(t)}}$ for each partition. Then, the sampling step draws an assignment with corresponding partition. This is done by building the cumulative distribution function of the $\mathcal{L}^{\mathscr{p}_\alpha^{(t)}}$'s and drawing a uniform distributed random number that samples the assignment $\tilde{\boldsymbol{\varphi}}_\alpha^{(t)}$ with corresponding partition and components $\mathcal{C}_{k|k}^{(\alpha)}$. These and the partition weight are added to the set $\boldsymbol{\Phi}_k$ (line 14). For the next iteration $t+1$, the assignment denotes $\boldsymbol{\varphi}^{(t+1)} = \tilde{\boldsymbol{\varphi}}_\alpha^{(t)}$. The main loop runs for T times and in the end, the partitions with highest weights $\mathcal{W}_{\mathscr{p}}$ of set Φ_k can be processed in the kth iteration of the filter loop, see Table 13.

Table 14: Stochastic Partitioning step

Given: Measurement set $\boldsymbol{Z}_k$, iteration factor T_{StP}, number of predicted components $J_{k|k-1}$.

Initialize:

```
1:  $T = T_{StP} \cdot (J_{k|k-1} + |\boldsymbol{Z}_k|)$ # number of sampling iterations
2:  Set initial assignment vector $\boldsymbol{\varphi}^{(t)}$
3:  $\mathcal{C}_{k|k} = correction\_step\left(\mathcal{C}_{k|k-1},\ \mathscr{p}\big(\boldsymbol{\varphi}^{(t)}\big)\right)$
4:  $\boldsymbol{\Phi}_k = \emptyset$ # empty set for drawn assignments, GIW components and partition weights in each iteration
```

Main loop:

```
5:  for $t = 1, \dots, T$
6:      do: Stochastic Partitioning actions for $\boldsymbol{\varphi}^{(t)}$ according to Table 12
7:      result: $n_\alpha$ different $\boldsymbol{\varphi}_\alpha^{(t)}$ with corresponding $\mathscr{p}_\alpha^{(t)}$
8:      for $\alpha = 1, \dots, n_\alpha$
9:          $\mathcal{C}_{k|k}^{(\alpha)} = partial\_correction\_step\left(\mathcal{C}_{k|k},\ \mathcal{C}_{k|k-1},\ \boldsymbol{W}_{c_1}^{\mathscr{p}_\alpha^{(t)}},\ \boldsymbol{W}_{c_2}^{\mathscr{p}_\alpha^{(t)}}\right)$
10:         compute partition weight $\mathcal{W}_{\mathscr{p}}^{(\alpha)} = \prod_{\boldsymbol{W} \in \mathscr{p}_\alpha^{(t)}} d_{\boldsymbol{W}}$
11:     end for
12:     compute relative likelihood $\mathcal{L}^{\mathscr{p}_\alpha^{(t)}} = \frac{\mathcal{W}_{\mathscr{p}}^{(\alpha)}}{\sum_{\alpha'} \mathcal{W}_{\mathscr{p}}^{(\alpha')}}$ for each partition $\mathscr{p}_\alpha^{(t)}$
13:     sample assignment $\tilde{\boldsymbol{\varphi}}_\alpha^{(t)}$ with acceptance probability $P\left(\boldsymbol{\varphi}^{(t+1)} = \boldsymbol{\varphi}_\alpha^{(t)} \middle| \boldsymbol{\varphi}^{(t)}\right) = \mathcal{L}^{\mathscr{p}_\alpha^{(t)}}$
14:     add partition $\mathscr{p}^{(t)} = \mathscr{p}\left(\tilde{\boldsymbol{\varphi}}_\alpha^{(t)}\right)$ with corresponding $\mathcal{W}_{\mathscr{p}}^{(t)}$ and $\mathcal{C}_{k|k}^{(t)}$ to set $\boldsymbol{\Phi}_k$
15:     set assignment for next iteration $\boldsymbol{\varphi}^{(t+1)} = \tilde{\boldsymbol{\varphi}}_\alpha^{(t)}$
16: end for
```

```
17: take the $L$[19] highest weights $\mathcal{W}_{\wp}$ of set $\boldsymbol{\Phi}_k$ and extract the corresponding corrected filter
        components and weights.
```

Output: Set of GIW mixture components $\left\{w_{k|k}^{(n)}, \xi_{k|k}^{(n)}\right\}_{n=1}^{J_{k|k}} \subseteq \{\mathcal{C}\}_{n=1}^{J_{k|k}} \in \boldsymbol{\Phi}_k$

The StP algorithm in Table 14 requires the possibility of a partial correction (line 9), i.e. the correction of GIW mixture components that are dependent on certain cells $\boldsymbol{W}$, while all components depending on other cells stay unchanged. In every iteration t up to $n_\alpha = 2|\mathbb{S}^t| + 1$ new partitions are obtained, cf. (6.27). To obtain the weight (line 10, Table 14), the cell weights $d_{\boldsymbol{W}}$ are necessary. One way to do this, is the computation of n_α times the complete correction step, which is computationally expensive. The other and computationally faster way is, to compute a complete correction of the GIW mixture components only for the initial partition (line 3) and a partial correction for all following cells (line 9), that are changed due to the StP actions. The partial correction affects only the GIW components depending on the cells $\boldsymbol{W}_{c_1}^{(t)}$ and $\boldsymbol{W}_{c_2}^{(t)}$, which are the only cells that are changed during the StP actions, cf. Table 12. Then, the weight can be computed with much less computational effort, which will be exemplified now:

The unnormalised weight of a partition is given in (6.29) as $\mathcal{W}_{\wp} = \prod_{\boldsymbol{W}\in\wp} d_{\boldsymbol{W}}$. Assuming at iteration t of the StP algorithm, there is the initial assignment $\boldsymbol{\varphi}^{(t)}$ with corresponding partition $\wp^{(t)} = \wp(\boldsymbol{\varphi}^{(t)})$ and weight $\mathcal{W}_{\wp}^{(t)}$. Applying e.g. an action of type 2 of Table 12, an assignment vector $\boldsymbol{\varphi}_\alpha^{(t)}$ with corresponding partition $\wp_\alpha^{(t)}$ is obtained, where only the cells $\boldsymbol{W}_{c_1}^{(t)}$ and $\boldsymbol{W}_{c_2}^{(t)}$ are changed so that the resulting weight $\mathcal{W}_{\wp}^{(\alpha)}$ can be calculated using

$$\mathcal{W}_{\wp}^{(\alpha)} = \mathcal{W}_{\wp}^{(t)} d_{\boldsymbol{W}_{c_1}}^{\wp_\alpha^{(t)}} \left(d_{\boldsymbol{W}_{c_1}}^{\wp^{(t)}}\right)^{-1} d_{\boldsymbol{W}_{c_2}}^{\wp_\alpha^{(t)}} \left(d_{\boldsymbol{W}_{c_2}}^{\wp^{(t)}}\right)^{-1}. \tag{7.1}$$

It can be seen that the weight of a partition obtained by an StP action, can be easily calculated using the weight of the initial partition $\mathcal{W}_{\wp}^{(t)}$ multiplied with the weights of the affected cells. In (7.1) $d_{\boldsymbol{W}_{(\cdot)}}^{\wp_\alpha^{(t)}}$ denotes the cell weight of the cell after applying an action to it and $d_{\boldsymbol{W}_{(\cdot)}}^{\wp^{(t)}}$ denotes the cell weight before applying an action. For the definition of the cell weight $d_{\boldsymbol{W}}$, see Table 6. For the general case, the partition weight after applying an action is expressed as

$$\mathcal{W}_{\wp}^{(\alpha)} = \mathcal{W}_{\wp}^{(t)} \mathcal{W}_{\alpha}^{(t)}, \tag{7.2}$$

where $\mathcal{W}_{\alpha}^{(t)}$ denotes the product of the affected cell weights. Further, the computation of the relative likelihood $\mathcal{L}^{\wp_\alpha^{(t)}}$ can now be simplified inserting (7.2) into (6.30), which leads to

$$\mathcal{L}^{\wp_\alpha^{(t)}} = \frac{\mathcal{W}_{\alpha}^{(t)}}{\sum_{\alpha'} \mathcal{W}_{\alpha'}^{(t)}}. \tag{7.3}$$

For the implementation of the StP method, the correction step of the classical GIW-PHD framework of Table 7 has to be modified to make it possible to change single GIW components after an initial correction.

To clarify the problems of the classical correction step, the pseudo code for the GIW-PHD correction step, which is provided completely in [43, p. 3], is given as simplified pattern in Table 15.

[19] In the practical implementation, $L = 5$ was used.

Table 15: Classical GIW-PHD correction pattern

```
Given: predicted GIW mixture components $\mathcal{C}_{k|k-1} = \left\{w_{k|k-1}^{(j)}, \xi_{k|k-1}^{(j)}, \dots\right\}_{j=1}^{J_{k|k-1}}$ and partitions $\left\{\wp^{(p)}\right\}_{p=1}^{P}$
1:  $l = 0$
2:  for $p = 1, \dots, P$ # loop over every partition
3:      for $w = 1, \dots, \left|\wp^{(p)}\right|$ # loop over every cell in $p$th partition
4:          $l = l + 1$ # cell counter
4:          for $j = 1, \dots, J_{k|k-1}$ # loop over every predicted component
5:              $\mathcal{C}_{k|k}^{(j+J_{k|k-1}\cdot l)} \propto \mathcal{C}_{k|k-1}, \boldsymbol{W}_{w}^{(p)}$ # components correction dependent on the cells
6:          end for
7:      end for
8:  end for
Output: updated GIW mixture components $\mathcal{C}_{k|k} = \left\{w_{k|k}^{(j)}, \xi_{k|k}^{(j)}\right\}_{j=1}^{J_{k|k}}$
```

The corrected components $\mathcal{C}_{k|k}^{(\cdot)}$ are obtained in the correction step over three for-loops with l as a cell counter, which is incorporated in the indexing $j + J_{k|k-1} \cdot l$. This method of indexing leads to a loss of information. After the classical correction step, it is not possible to change a single components $\mathcal{C}_{k|k}^{(j)}$ because with the indexing it is not obvious, which cell $\boldsymbol{W}_{w}^{(\wp)}$ was used for the correction. But this is needed for StP to obtain a faster algorithm. StP changes only single cells due to the actions and therefore the *partial_correction_step*$(\cdot)$ was implemented and the *correction_step*$(\cdot)$ for StP was modified. The problem can be solved by changing the indexing of the *correction_step*$(\cdot)$ in a such way, that it becomes dependent on the cells of the certain partitions. Applying this, the corrected components (line 5, Table 15) can be subscripted as

$$\mathcal{C}_{k|k}^{(p,w,j)} \propto \mathcal{C}_{k|k-1}, \boldsymbol{W}_{w}^{(p)}, \tag{7.4}$$

where p is the partition counter, w the cell counter and j the jth predicted component, cf. Table 15. With this new indexing, the additional function *partial_correction_step*$(\cdot)$ can be implemented. This function processes the corrected components $\mathcal{C}_{k|k}$ with the new indexing, the predicted components $\mathcal{C}_{k|k-1}$ and the affected cells $\boldsymbol{W}_{c_1}^{\wp_\alpha^{(t)}}$ and $\boldsymbol{W}_{c_2}^{\wp_\alpha^{(t)}}$ to do a partial correction, denoted as

$$\mathcal{C}_{k|k}^{\left(\wp_\alpha^{(t)}, c_1, j\right)} \propto \mathcal{C}_{k|k-1}, \boldsymbol{W}_{c_1}^{\wp_\alpha^{(t)}} \text{ with } j = 1, \dots, J_{k|k-1} \tag{7.5}$$

and

$$\mathcal{C}_{k|k}^{\left(\wp_\alpha^{(t)}, c_2, j\right)} \propto \mathcal{C}_{k|k-1}, \boldsymbol{W}_{c_2}^{\wp_\alpha^{(t)}} \text{ with } j = 1, \dots, J_{k|k-1}. \tag{7.6}$$

These suggestions should simplify the implementation and are not further illustrated here. With the GIW mixture components that are stored dependent on the cells of a partition, it becomes possible to change only single components in the partial correction step of Table 14.

7.3. Comments on Stochastic Partitioning

The presented sampling algorithm in section 7.2 has shown better performance in simulations than classical heuristic partitioning methods. For a further discussion and simulation results see chapter 8. However, some comments are given here that show a major drawback and some issues of the implementation.

The drawback that was brought up during the simulation runs of the GIW-PHD filter with StP was, that the algorithm is highly dependent on the prediction step. This fact can be easily explained: The sampling algorithm samples partitions using the predicted GIW mixture components, but if the prediction is poorly computed – means the motion model is badly chosen to the application – the filter diverges for the remaining observation time. In comparison with a heuristic approach like DP, this property is not existent as long as the partitioning approach is independent of the prediction step.

Further, when computing weights as the partition weights, cf. Table 14, or the components weight in the correction step, given in Table 6, the numbers often lead to a numerical overflow. Knowing this, the implementation can be done using the logarithm of the weights as proposed in [43, p. 5].

8. Simulation results

This chapter presents the results of the implemented filters, which were tested in simulated tracking scenarios. One implementation was done using the originally proposed GIW-PHD filter implementation of Granström in [6] [43] with the classical partitioning methods DP and SP. The second implementation consists of the adapted GIW-PHD filter framework that was presented in chapter 7. The adapted GIW-PHD filter uses the proposed sampling method Stochastic Partitioning of chapter 6. Both implementations were tested during the implementation with a simulation environment creating random scenarios (SECRS), where elliptical objects are born, die and spawn. Further, a typical scenario of two objects, which approach each other and move in parallel before separating, is used to perform Monte Carlo simulations to assess the different partitioning methods.

In section 8.1, the analysis techniques for the performance evaluation are defined. The OSPA (Optimal Sub-Pattern Assignment) and H-OSPA miss distances are introduced. Section 8.2 presents the simulation environment that is used to test the implementations. Section 8.3 presents the functionality test of the filters and in section 8.4, the results of the Monte Carlo simulations are presented.

8.1. Performance analysis techniques

The GIW-PHD filter is a multi-object tracker that processes extended objects and therefore its main task is to determine the number of objects in a scenario and the objects' state. In the analysis of the Monte Carlo simulations, different aspects are considered separately:

- The number of objects over time, i.e. the cardinality, is evaluated using the sum of weights and the total number of extracted objects.
- The extent estimation is analysed comparing the true size of the major and minor half-axes of the ellipses with the estimated ones by the filter.
- To compare the computational effort of the different partitioning methods, the absolute execution times are analysed and compared among each other. All different Monte Carlo simulations where done on the same computer system and the execution times are averaged values of 1000 simulation runs.

To obtain a reasonable performance comparison between the filters with its different partitioning methods, the multi-object metric OSPA is used as a miss distance measure. The OSPA is a consistent metric for the performance evaluation of multi-object filters and was presented in [59]. The general problem of performance evaluation in multi-object scenarios is given in Figure 3. In (A) the cardinality is estimated correctly, but the single distances between the true objects and estimates are large. Whereas in (B), one estimate is close to one true object but the second object is missed, i.e. the cardinality is underestimated. In (C), two estimates are close to the true objects but the third estimate is a false estimate, leading to a cardinality overestimation.

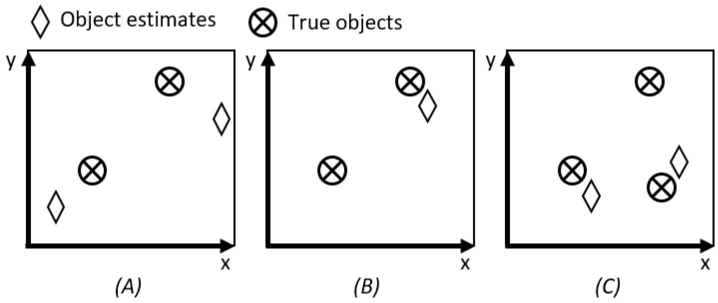

Figure 3: Paradigmatic multi-object scenarios. It is difficult to evaluate, which of the three scenarios is closest to the truth.

The OSPA is a metric defined on the space of finite sets and captures cardinality and position state errors meaningfully. For a filter comparison the advantage is, that the OSPA generates one number, which can be easily compared with other filter variants. If the number increases, the filter performance becomes worse, if the number decreased, the filter performance improves – this behaviour leads the physical interpretation to be intuitive.

Assuming two sets $\boldsymbol{X} = \{\boldsymbol{x}_1, \dots \boldsymbol{x}_m\}$ and $\boldsymbol{Y} = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_n\}$ on an equal space $\mathcal{Y}$, the OSPA evaluates the miss distance between both sets using a metric $d^{(c)}$, which denotes the distance between $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{Y}$ with a cut off parameter $c > 0$:

$$d^{(c)}(\boldsymbol{x}, \boldsymbol{y}) \triangleq \min(c, d(\boldsymbol{x}, \boldsymbol{y})) \tag{8.1}$$

Further, let Π_k be the set of permutations on $\{1,2,3, \dots, k\}$ for any non-negative integer $k \in \mathbb{N} = \{1,2,3, \dots\}$, then the OSPA of the two finite sets $\boldsymbol{X}$ and $\boldsymbol{Y}$ is defined as [59]

$$\bar{d}_p^{(c)}(\boldsymbol{X}, \boldsymbol{Y}) = \left(\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^{m} d^{(c)} \left(\boldsymbol{x}_i, \boldsymbol{y}_{\pi(i)} \right)^p + c^p (n - m) \right) \right)^{\frac{1}{p}}, m \leq n. \tag{8.2}$$

If $m > n$, the OSPA is defined as $\bar{d}_p^{(c)}(\boldsymbol{X}, \boldsymbol{Y}) \triangleq \bar{d}_p^{(c)}(\boldsymbol{Y}, \boldsymbol{X})$. In (8.2) the parameter c denotes the cut off and p is the order of the metric. For $p = 2$ the distance measure is the Euclidean distance. Further, the cut off is set to $d = 60$, as this is the same choice in [6]. Within the filter performance evaluation, the space $\mathcal{Y}$ is the measurement space, which leads to $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^2$. For the filter evaluation the sets $\boldsymbol{X}$ and $\boldsymbol{Y}$ are set to the estimated state set and the true object state set, whereas the state sets only contain the x, y-coordinates of the objects. This implies that the OSPA captures the localization errors related on the centre of the elliptical extended objects. To include the extent estimation into the OSPA, a different measure can be used. In [60], the OSPA was extended with the Hellinger distance to incorporate an uncertainty in form of a covariance for the position estimation. This application is suitable for evaluating the objects extents in the GIW-PHD framework. With the random matrix approach, elliptically shaped objects represented by the covariance of a Gaussian distribution are obtained. The Hellinger distance d_H is a measure that is defined on two probability distribution f and g, given as [60]

$$d_H(f, g) = \frac{1}{2} \int \left(\sqrt{f(\boldsymbol{x})} - \sqrt{g(\boldsymbol{x})} \right)^2 d\boldsymbol{x}. \tag{8.3}$$

The range of values is given with $0 \leq d_H \leq 1$. A zero claims two identical shaped distributions that perfectly overlap and a one is obtained when there is no overlap. For the case of two Gaussian distributions, as it occurs using the random matrix framework, there is a closed form expression of the Hellinger distance, given as

$$\begin{gathered} d_H(f, g) = 1 - \sqrt{\frac{\sqrt{\det\left(\Sigma_x \Sigma_y\right)}}{\det\left(\frac{1}{2}\left(\Sigma_x + \Sigma_y\right)\right)} \cdot e^{\epsilon}} \\ \epsilon = \left(-\frac{1}{4} (\boldsymbol{x} - \boldsymbol{y})^T \left(\Sigma_x + \Sigma_y\right)^{-1} (\boldsymbol{x} - \boldsymbol{y}) \right). \end{gathered} \tag{8.4}$$

In case of the random matrix framework, the covariance matrices $\Sigma_{\boldsymbol{x}}$ and $\Sigma_{\boldsymbol{y}}$ correspond to the true extent matrix X_k and the estimated extent matrix $\hat{X}_k$ respectively. $\boldsymbol{x}$ denotes the true position of a simulated object and and $\boldsymbol{y}$ denotes the estimated position by the filter. The estimated position is defined with the first two elements of $\boldsymbol{m}_k$ after the state extraction, cf. Table

26. The so called H-OSPA results, inserting the Hellinger distance of (8.4) into (8.2). The H-OSPA is used to evaluate the filter implementations with a cut off parameter of $c = 1$.

8.2. Simulation environment

To test the functionality of the filter and its different partitioning methods, a SECRS was implemented. The SECRS creates complex multi-object scenarios that were used to evaluate the filters during the implementation and testing phase. For the scientific analysis of the implementations, a defined scenario is needed that can be used for Monte Carlos simulations. The implemented scenario is called parallel track with approaching. It consists of two objects that approach each other and come close up to a defined distance. Then the objects move in parallel for a certain time before they separate. This scenario was chosen to capture the difficult situation for partitioning methods, when objects approach each other and move parallel with small distances.

8.2.1. Random scenarios

The SECRS defines a half circle as observation area with a radius of $500m$. Outside the observation area, objects can move but do not create measurements. The surveillance area of a randomly created run is illustrated in Figure 4.

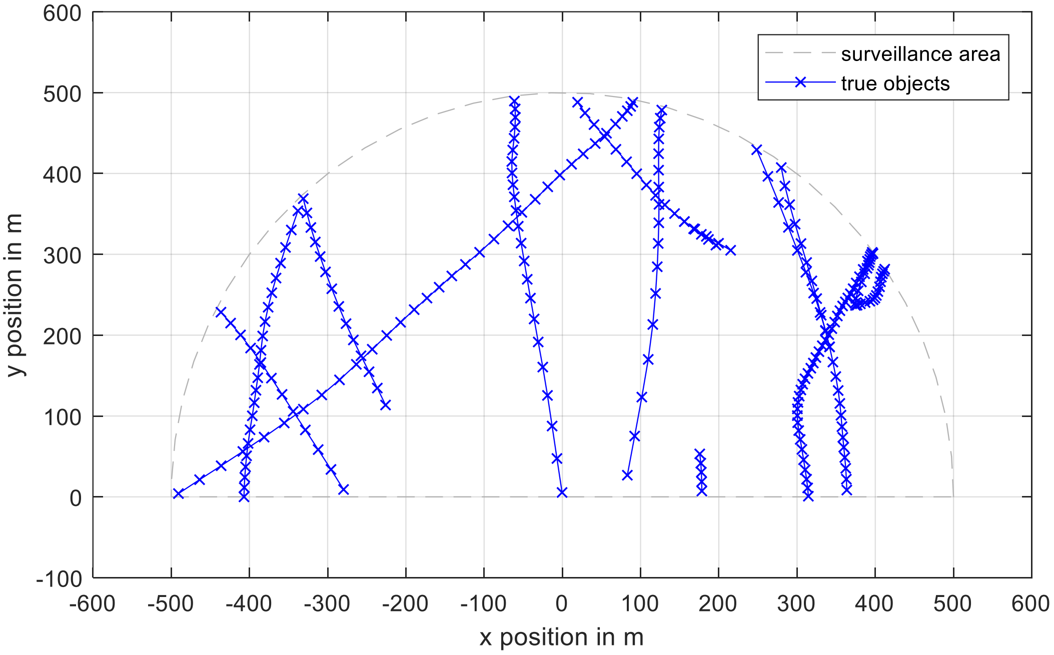

Figure 4: An example of a random multi-object scenario created with SECRS. The blue lines are the objects' trajectories over an observation time of 50s. Note that the object extent is not illustrated.

The surveillance area is illustrated as grey dashed line and the objects' trajectories as blue lines, where the crosses denote the sampling points.
The ith true object state is given as $\xi_k^{(i)} = (\boldsymbol{x}_k^{(i)}, X_k^{(i)})$, where the kinematics state denotes $\boldsymbol{x}_k^{(i)} = \left(x, y, v_x, v_y, a_x, a_y\right)_k^T$ and the random matrix is chosen as $X_k^{(i)} = q \cdot diag(50{,}10)$, where q is a discrete uniform distributed number $\in \{1{,}2, \dots, 5\}$ and the operation $diag(\cdot)$ denotes the diagonal matrix of the input arguments. In SECRS the number of initial objects n_{ξ_0} is selectable. The initial objects are randomly generated a little outside the surveillance area with a start velocity $\boldsymbol{v}_0 = 22.22m \cdot s^{-1}$ $(80km \cdot h^{-1})$. The velocity $\boldsymbol{v}_0$ is directed into the inside of the surveillance

area. The ith extension matrix is rotated around the angle ϕ so that the major axis is aligned to the velocity direction using the matrix rotation

$$X_k^{(i)} = R_k^{(i)} X_k^{(i)} \left(R_k^{(i)}\right)^T \tag{8.5}$$

with $R_k^{(i)} = \begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix}$. In the SECRS, object actions are implemented that allow the the birth of new objects outside the surveillance area with a probability $p_\gamma = 0.05$, and the spawning[20] of objects with $p_\beta = 0.03$. When a child object is spawned, it is duplicated on the same position with the same extent as its spawn father. Further, the simulation environment can merge objects that hit each other. A hit is defined, when the major half-axes in heading direction cross each other. The resulting object is the average of the merged objects. Further, the objects have a survival probability of $p_S = 0.99$ and therefore can die at every time step with the probability of $1 - p_S$.

Each objects' motion and measurements are given according to Koch's models described in chapter 4. The corresponding motion model parameters for the tests with SECRS are listed in Table 16, whereas the measurement model parameters are given in Table 17.

The measurement rate γ_k is set constant and independently of the objects' extent. For a simple measurement rate model that involves the extent matrix X_k, the reader is referred to [6]. To obtain a measurement rate that is additionally dependent on the object's distance to the sensor, see [9]. Further, the clutter measurements are uniformly distributed over the surveillance area with the false alarm rate $\beta_{FA,k}$ which leads for a surveillance volume of $V_S = \frac{\pi}{2} 500^2 m^2$ to ten clutter measurements per time step.

Table 16: SECRS motion model parameters

Sampling time	T_S	$0.5s$
Manoeuvre correlation time	θ	$1.0s$
Acceleration standard deviation	Σ	$0.5m \cdot s^{-2}$

Table 17: SECRS measurement model parameters

Measurement rate	γ_k	10
False alarm rate	$\beta_{FA,k}$	$\approx 2.465 \cdot 10^{-5} / V_S$
Mean number of clutter	$\lambda_{FA,k}$	10
Detection probability	p_D	0.99

[20] Spawning of objects is not modelled explicitly in the GIW-PHD filter implementation but came out to be covered implicitly by the heuristic partitioning methods, see section 8.3.

8.2.2. Parallel track with approaching

The scenario used for the Monte Carlo simulations is illustrated in Figure 5. Two objects start with separated positions and approach each other up to a certain distance, move parallel and finally separate again.

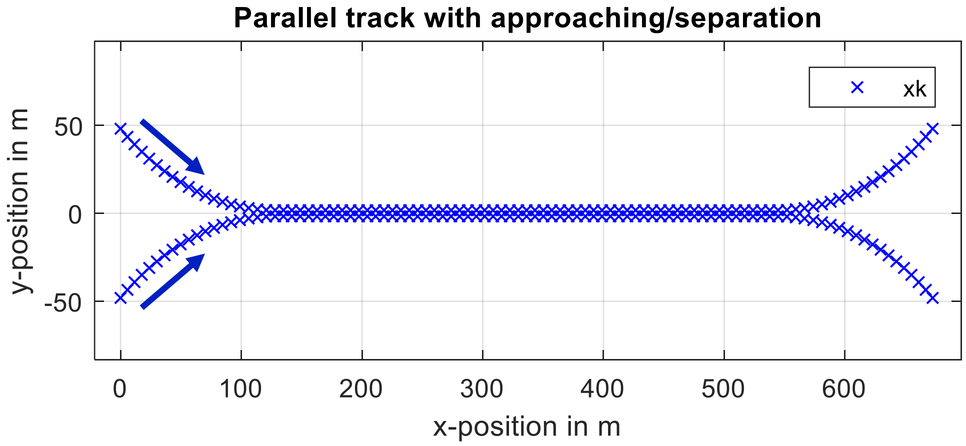

Figure 5: Parallel track with approaching and separation. The arrows indicate the objects' direction of motion. Note that the objects' extent is not illustrated.

The two objects in this scenario have different sizes. The major half-axes Ai and minor half-axes ai of the ith object are given as

$$\begin{aligned} A1 &= 5m, \\ a1 &= 2m, \\ A2 &= 2.5m, \\ a2 &= 1m, \end{aligned} \tag{8.6}$$

whereas the distance between the objects is given in relation to the 3-sigma ellipse of the object's extent. Thus, the random matrix $X_k^{(i)}$ is defined as $X_k^{(i)} = diag\left(\left(\frac{Ai}{3}\right)^2, \left(\frac{ai}{3}\right)^2\right)$. In the Monte Carlo simulations, the object distances of $0.5m$, $0.25m$ and $0.0m$ were applied. The objects move with a constant velocity of $\boldsymbol{v}_0 = 13.89m \cdot s^{-1}$ $(50km \cdot h^{-1})$ on the given trajectory, whereas the rotation of the extension matrix is performed as described in (8.5). The objects' measurements are given according to Koch's models described in chapter 4 with the corresponding parameters given in Table 18. Clutter measurements are uniformly distributed over the surveillance area. The surveillance area is defined in x-direction from $-100m$ to $900m$ and in y-direction from $-250m$ to $250m$, with a resulting surveillance volume of $V_S = 5 \cdot 10^5 m^2$.

Table 18: Parallel tracks measurement model parameter

Measurement rate	γ_k	10
False alarm rate	$\beta_{FA,k}$	$\approx 6.1625 \cdot 10^{-5}/V_S$
Mean number of clutter	$\lambda_{FA,k}$	31
Detection probability	p_D	0.99

8.3. Functionality test with SECRS

In this section, the filter is tested with the different partitioning methods using the SECRS as described in subsection 8.2.1. For DP and StP an exemplary simulation run is presented with comments on the typical characteristics of the respective partitioning method. The filter parameters for the simulation run are given in Table 19.

Table 19: PHD filter parameters for SECRS tests

Number of initial objects	n_{ξ_0}	5
Number of birth PHD mixtures	$J_{\gamma,k}$	30
Weight of birth PHD mixture	$w_{\gamma,k}$	$1/J_{\gamma,k}$
Birth inverse scale matrix	$V_{\gamma,k}^{(j)}$	$diag(250, 50)$
Birth degrees of freedom	$\nu_{\gamma}^{(j)}$	7
Extent decay constant	τ	5
Manoeuvre correlation time	θ	$1s$
Pruning truncation threshold	T	10^{-5}
Merging threshold	U	4
Maximum number of components	J_{max}	100
Extraction weight threshold	$\hat{w}$	0.5

The birth PHD mixture components are defined with the means $\boldsymbol{m}_{\gamma,k}^{(j)}$ and the covariance matrices $P_{\gamma,k}^{(j)}$. The components are equally spaced over the edge of the surveillance area as shown in Figure 6. The means of the birth PHD are given as

$$\boldsymbol{m}_{\gamma,k}^{(j)} = \left(\boldsymbol{x}_0^{(j)}, \boldsymbol{v}_0^{(j)}, \boldsymbol{0}\right)^{\mathrm{T}}, \tag{8.7}$$

where $\boldsymbol{x}_0^{(j)}$ are the centres of the orange circles in Figure 6. The modulus of the velocity vector $\boldsymbol{v}_0^{(j)}$ is set to the start velocity of the true objects. The velocity vector is aligned towards the inside of the surveillance area and is set equal with the orientation of secant in the point $\boldsymbol{x}_0^{(j)}$. The covariance matrices are given as

$$P_{\gamma,k}^{(j)} = diag\left(\sigma_{pos}^2, 25, \Sigma^2\right)^{(j)}, \qquad\qquad P_{\gamma,k}^{(j)}$$

where Σ is set to standard deviation of the true objects' acceleration, see Table 16. Further, ${\sigma_{pos}^2}^{(j)}$ denotes the variance for the position of the jth birth PHD component. ${\sigma_{pos}^2}^{(j)}$ is determined is such a way that the size of the one sigma area is obtained as given in Figure 6 with the orange circles.

The parameters of the partitioning methods DP and StP are given in Table 20. The set of thresholds for DP, cf. (6.20), contains $n_d = 171$ different distance thresholds when using the threshold parameters of Table 20. The total number of the iterations T for StP is given according to Table 14 as $T = T_{StP} \cdot (J_{k|k-1} + |\boldsymbol{Z}_k|)$.

Table 20: SECRS partitioning parameters of DP and StP

DP lower threshold	d_L	$3.5m$
DP upper threshold	d_U	$21m$
DP threshold set step size	Δd	$0.1m$
StP iteration factor	T_{StP}	5

An exemplary simulation run of the GIW-PHD filter using DP is given in Figure 7, where the object trajectories are given in blue and the object estimates in red. The complex scenario is handled in a satisfying way, because all object trajectories are captured by the filter. For all sample points of the true objects, indicated by the blue crosses, estimates are obtained, given as the red circles. One conspicuity is the existence of four false estimates, which are indicated by red circles without blue crosses. Sometimes these false estimates occur in areas, where closely spaced clutter measurements are clustered into one cell.

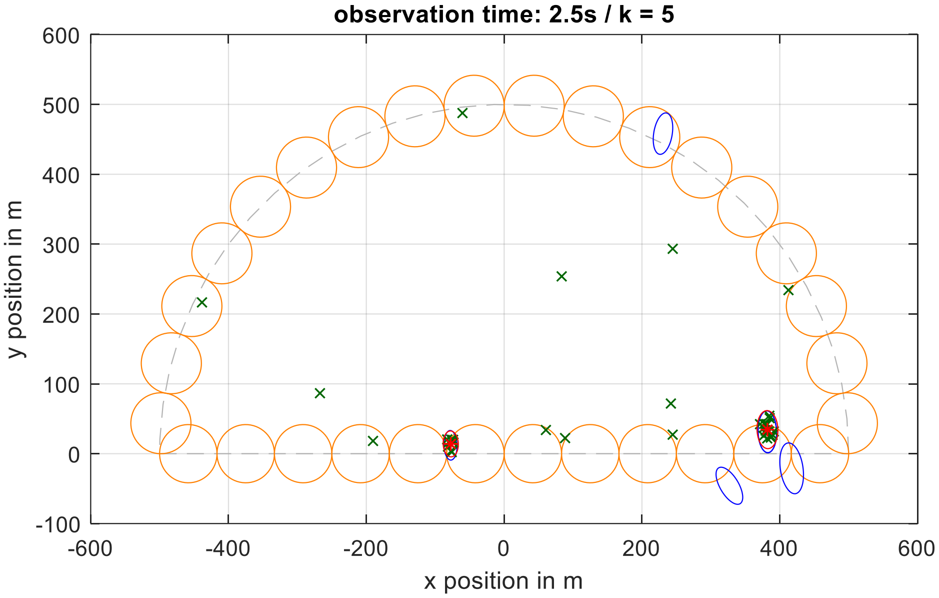

Figure 6: SECRS illustration of an exemplary time step with birth PHD (orange), true objects (blue), estimated objects (red) and measurement (green)

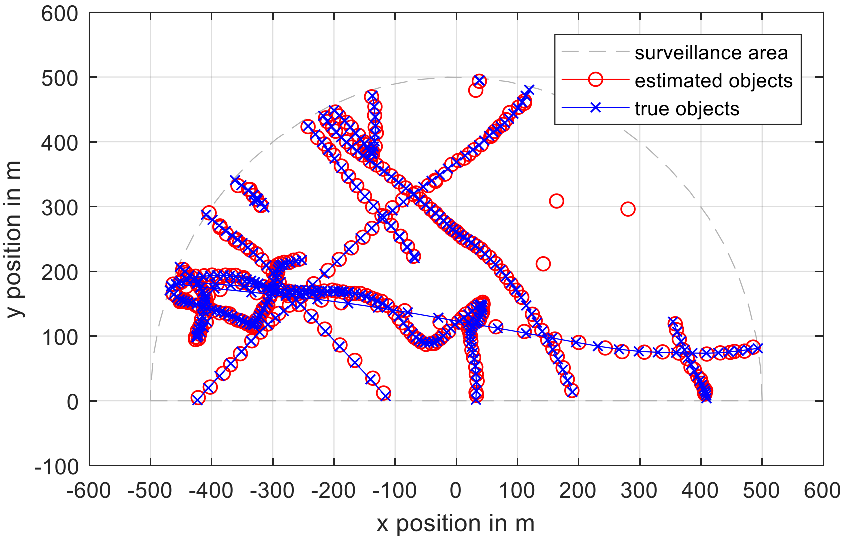

Figure 7: SECRS exemplary simulation run using DP

For the given scenario, the progress of the OSPA and the H-OSPA are given in Figure 8. Further, the trend of the cardinality is given in Figure 9. The figures show that the filter using DP performs an adequate cardinality output in most cases. The noise in the OSPA and H-OSPA is caused by to dynamic change of the object number due to birth, death and spawning processes. At time step $k = 58$, there is a cardinality underestimation that typically occurs, when an object spawns or two objects are spatially too close that DP is not capable of separating the measurement into several cells. In general, it can be stated that the filter with DP captures object spawning although this is not modelled explicitly in the filter equations, cf. (5.6). An explanation of this property is the heuristic partitioning method. When the objects are separated with a distance that can be captured by DP, those measurements are put into several cells leading the filter to create two object estimates.

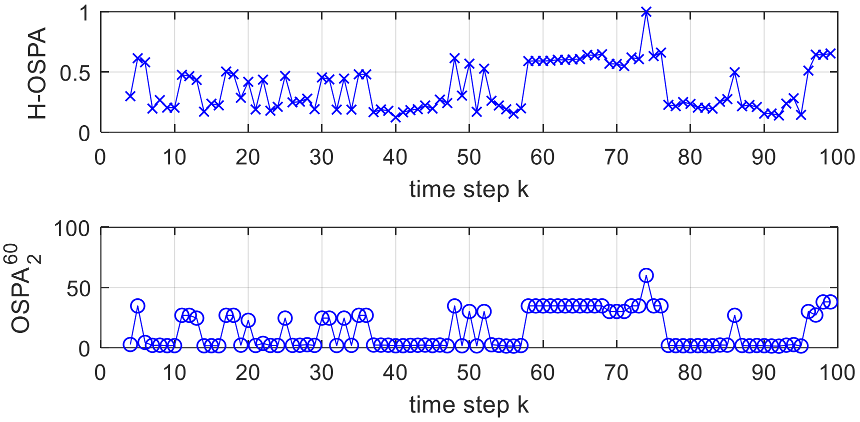

Figure 8: SECRS progress of the miss distances using DP

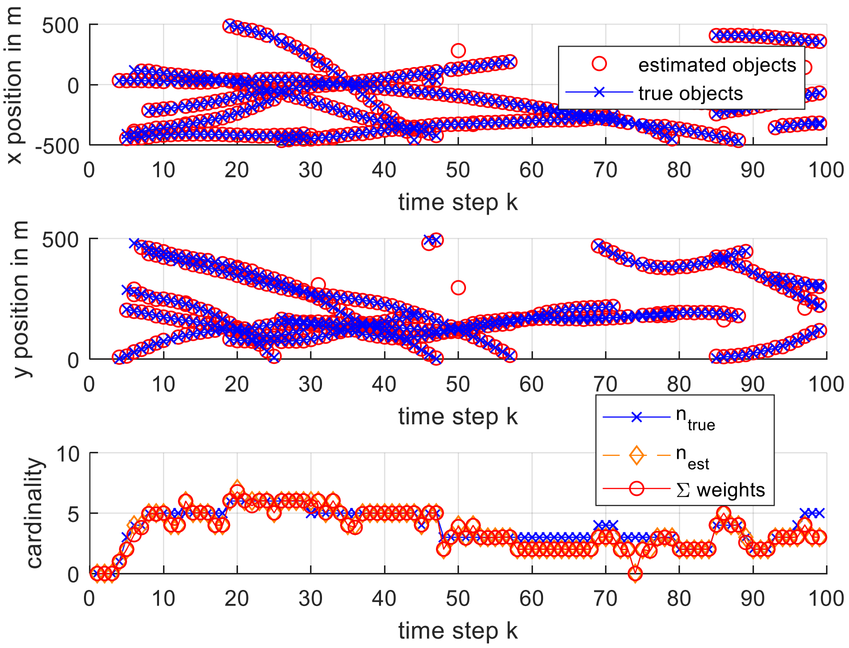

Figure 9: SECRS progress of the cardinality using DP. The true cardinality is given as blue line with markers 'x'. The red line with markers 'o' is the cardinality calculated as sum of the GIW mixture components' weights. The orange line with diamond marker is the number of estimated, i.e. extracted objects.

An exemplary simulation run of the GIW-PHD filter using StP is given in Figure 10. In comparison to the run using DP, the StP method has the conspicuity that objects, which enter the surveillance area, are not estimated during the first time step. There is a certain delay until the filter captures the objects. This is confirmed in Figure 12, where the cardinality follows the true number of objects with a certain delay.

An improvement compared to the use of DP is the absence of false estimates. Further, the miss distances in Figure 11 show less noise after time step $k = 28$, with one exception from $k = 47$, to $k = 62$, where the miss distances decrease because of a spawning event. Note that areas in the diagram without values denote the times, when no objects are existent in the surveillance area.

The spawning can better be seen in Figure 12. There, the first two subplots show the spawning event separated into x- and y-coordinates over the time steps. For the general case, spawning events using StP are captured as well but in a more unreliable way compared to DP. For scenarios where it is a priori known that spawning happens, this should be modelled explicitly, see e.g. [61].

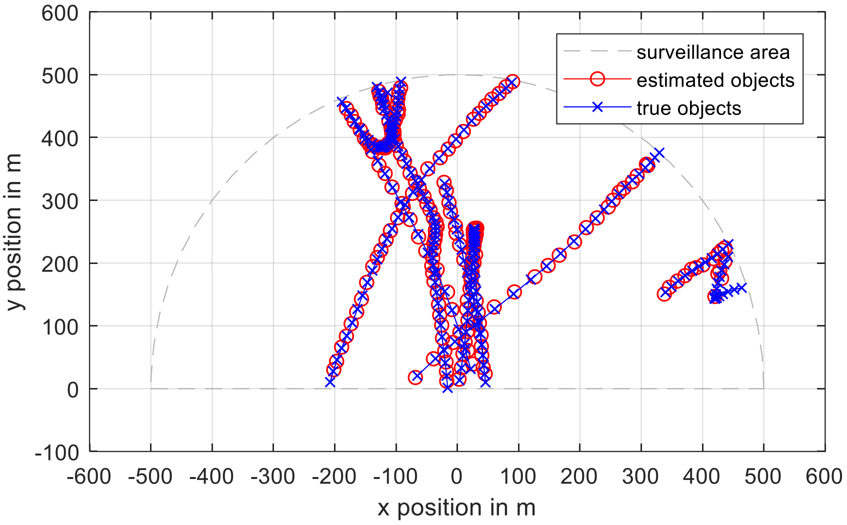

Figure 10: SECRS exemplary simulation run using StP

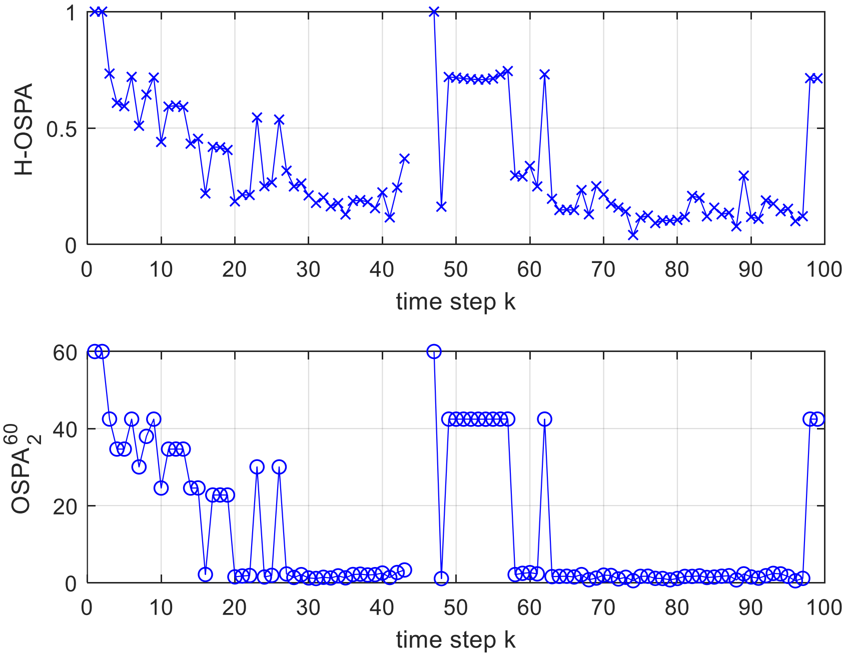

Figure 11: SECRS progress of the miss distances using StP

To conclude the functionality test using the SECRS, it can be stated that the filter using DP handles the birth of objects on the edge of the surveillance without time delay, whereas with the use of StP, the filter needs a certain number of time steps to create estimates. The death process is handled properly using both partitioning methods. Both methods can handle spawning, whereas the heuristic method DP performs more reliable. For StP, the spawning of objects should be modelled within the filter equations to perform in a reliable way. Further investigations should be done using Monte Carlo methods, to verify the given statements.

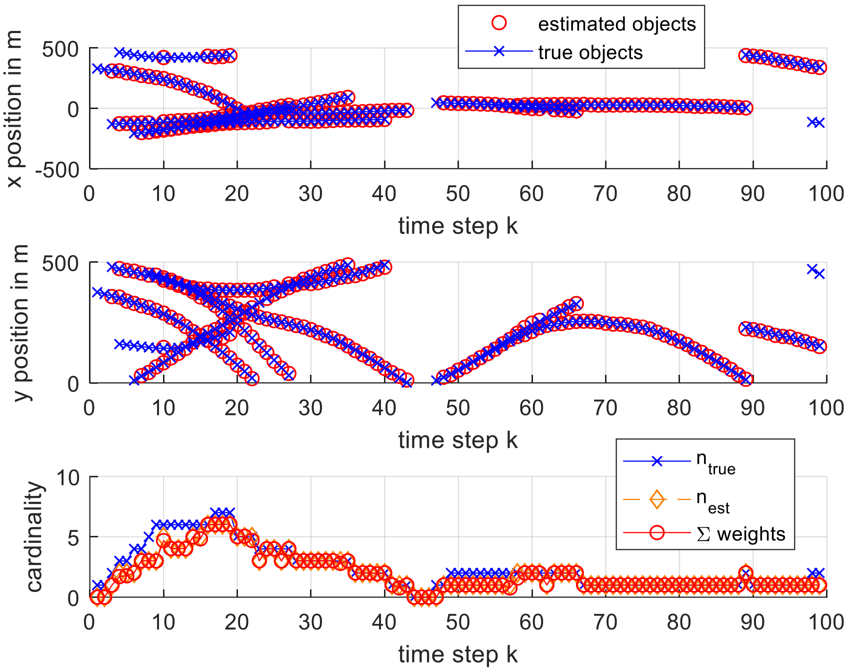

Figure 12: SECRS progress of the cardinality using StP. The true cardinality is given as blue line with markers 'x'. The red line with markers 'o' is the cardinality calculated as sum of the GIW mixture components' weights. The orange line with diamond marker is the number of estimated, i.e. extracted objects.

8.4. Performance evaluation of parallel tracks

In this section, the filter is applied to the parallel track scenario as described in subsection 8.2.2. This is done with different distances between the objects extent. The minimal distances between the objects were set to $0.5m$, $0.25m$ and $0.0m$. For each combination of the distances and the three partitioning methods, a Monte Carlo simulation with 1000 runs was performed. The three partitioning methods are given with DP, the combination of DP and SP, and StP. The results using the distance of $0.25m$ are given in the Appendix A.5. The filter parameters for the parallel track scenario are given in Table 21. The results of the Monte Carlo simulations are given in the subsections.

Table 21: PHD filter parameters for parallel track simulations

Number of birth PHD mixtures	$J_{\gamma,k}$	2
Weight of birth PHD mixtures	$w_{\gamma,k}$	$1/J_{\gamma,k}$
Birth inverse scale matrix	$V_{\gamma,k}^{(j)}$	$diag(100, 25)$
Birth degrees of freedom	$\nu_{\gamma}^{(j)}$	7
Decay constant, cf. Table 5	τ	$5s$
Manoeuvre correlation time	θ	$1s$
Pruning truncation threshold	T	10^{-5}
Merging threshold	U	4
Maximum number of components	J_{max}	100
Extraction weight threshold	$\widehat{w}$	0.5

The birth PHD's means and covariance matrices are set with the a priori information of the parallel track scenario. The means are set to $\boldsymbol{m}_{\gamma,k}^{(j)} = \left(\boldsymbol{x}_0^{(j)}, \boldsymbol{v}_0^{(j)}, \boldsymbol{0}\right)^{\mathrm{T}}$, where $\boldsymbol{x}_0^{(j)}$ is the starting position of the jth object according to Figure 5. $\boldsymbol{v}_0^{(j)}$ is set to object's true velocity, cf. subsection 8.2.2. The covariance matrices are given with $P_{\gamma,k}^{(j)} = diag(10{,}10{,}10)$ for both birth PHD components. The parameters of the partitioning methods DP, SP and StP are given in Table 22. The set of thresholds for DP contains $n_d = 73$ different distance. The total number of the iterations T for StP is given according to Table 14 as $T = T_{StP} \cdot (J_{k|k-1} + |\boldsymbol{Z}_k|)$.

Table 22: SECRS partitioning parameters of DP and StP

DP lower threshold	d_L	$0.5 \cdot a2 = 0.5m$
DP upper threshold	d_U	$3 \cdot A1 = 15m$
DP threshold set step size	Δd	$0.2m$
StP iteration factor	T_{StP}	5

The choice of the lower and upper threshold of DP can be reasoned by the extent of the two objects. The larger object has a major half axis of $A1 = 5m$. With the choice of d_U, it is ensured that even when one measurement is in the front part of the ellipsis and one in the rear part, there will be one partition where these measurements are clustered in one cell. The lower threshold ensures that the minimal spatial distance between the objects can be theoretically $0.5m$. The following results even show a proper performance for DP with lower object distances as $0.5m$.

8.4.1. Simulation results with a distance of $0.5m$

The OSPA and H-OSPA are given in Figure 13 for DP, in Figure 14 for the combination of DP and SP and in Figure 15 for StP. Due to the fact that the objects follow the given trajectory of Figure 5 with a constant velocity and without acceleration, the standard deviation of the acceleration was set to a small values with $\Sigma = 0.1m \cdot s^{-2}$. This first guess was used for all partitioning methods and directly influences the process noise covariance $Q_{k|k-1}$ (4.14). For DP and SP the MC simulations were successful but for StP the covariance had to be increased to avoid a filter divergence. Out of this reason, the scalar standard deviation of acceleration was set to $\Sigma = 0.5m \cdot s^{-2}$ for the use of StP, which leads the filter to a stable performance.

In Figure 13 and Figure 14 it can be seen that the partitioning methods DP and its combination with SP lead to high miss distances. This is the result of the of the cardinality underestimation, cf. Figure 16 and Figure 17, when the objects are spatially close at the simulation time of $10s$. In Figure 14 for the use of DP and SP, the H-OSPA shows a downward trend and further, the OSPA is lower compared to the single use of DP, cf. Figure 13. The H-OSPA distance with the use of StP, cf. Figure 15, converges to a stable and low level. The OSPA shows a nearly constant level at every time step. The cardinality estimates in Figure 18 reflect this trend. When comparing the OSPA and H-OSPA using StP, the H-OSPA starts on a high level which is the cause of the initialisation of the birth PHD components. Those are initialised larger than the true objects' extents, which leads the filter to need a certain time for performing accurate extension estimates. The H-OSPA converges distinctly below a value of 0.2. On the other hand, the OSPA using StP starts on a low level and increases a little for the duration when the two objects are close. The latter implies that the estimation of the objects' centroids is performed properly.

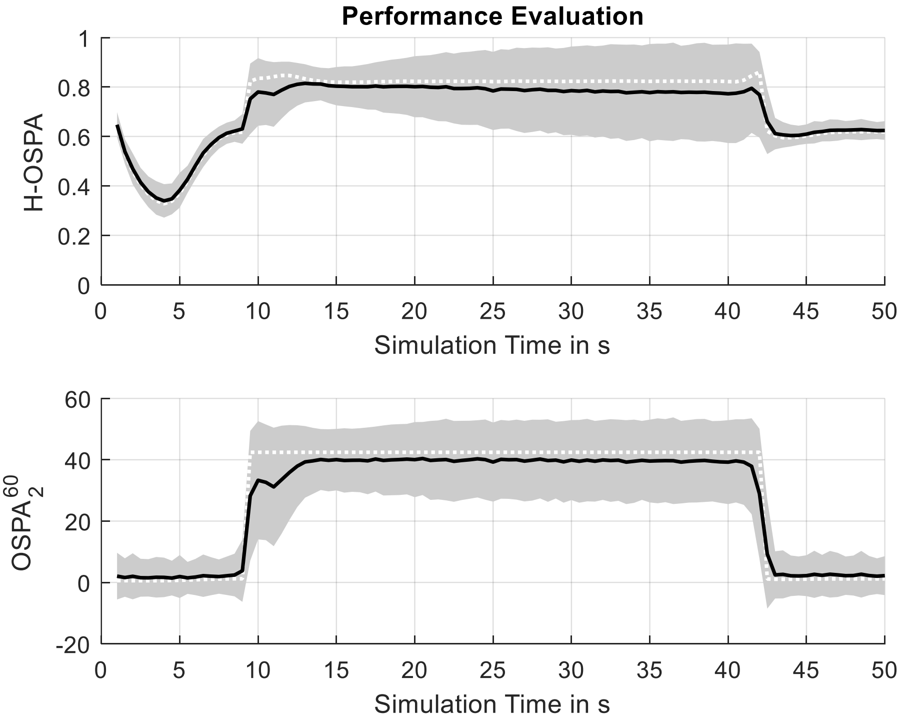

Figure 13: OSPA of DP at 0.5m distance between objects with $\Sigma = 0.1m \cdot s^{-2}$. The black line is the mean value with the grey area showing the +/- standard deviation range. The white dotted line gives the median value.

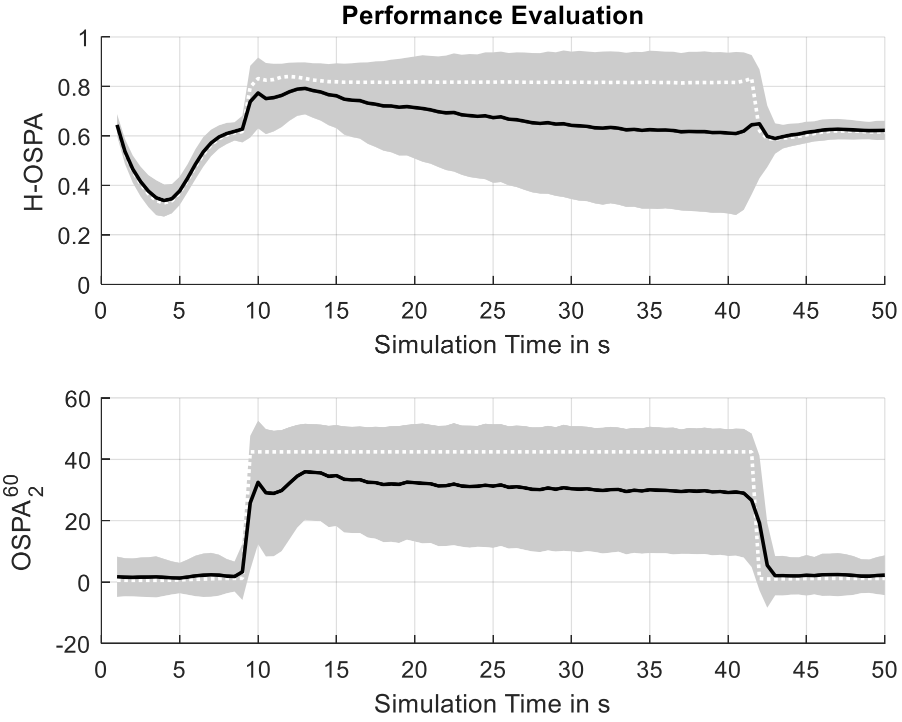

Figure 14: OSPA of DP with SP at 0.5m distance between objects with $\Sigma = 0.1m \cdot s^{-2}$. The black line is the mean value with the grey area showing the +/- standard deviation range. The white dotted line gives the median value.

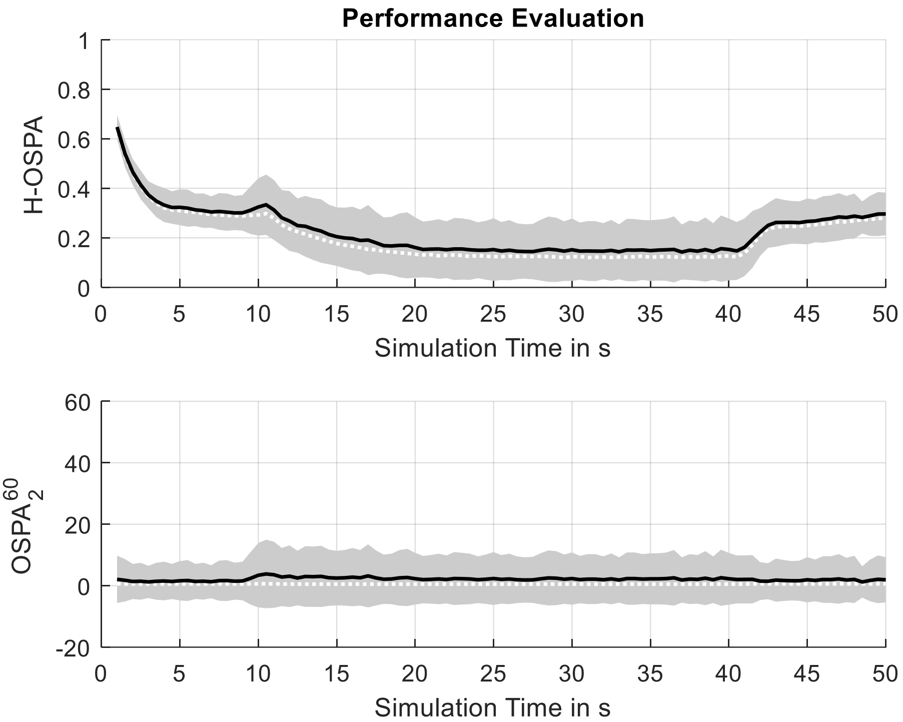

Figure 15: OSPA of StP at 0.5m distance between objects with $\Sigma = 0.5m \cdot s^{-2}$The black line is the mean value with the grey area showing the +/- standard deviation range. The white dotted line gives the median value.

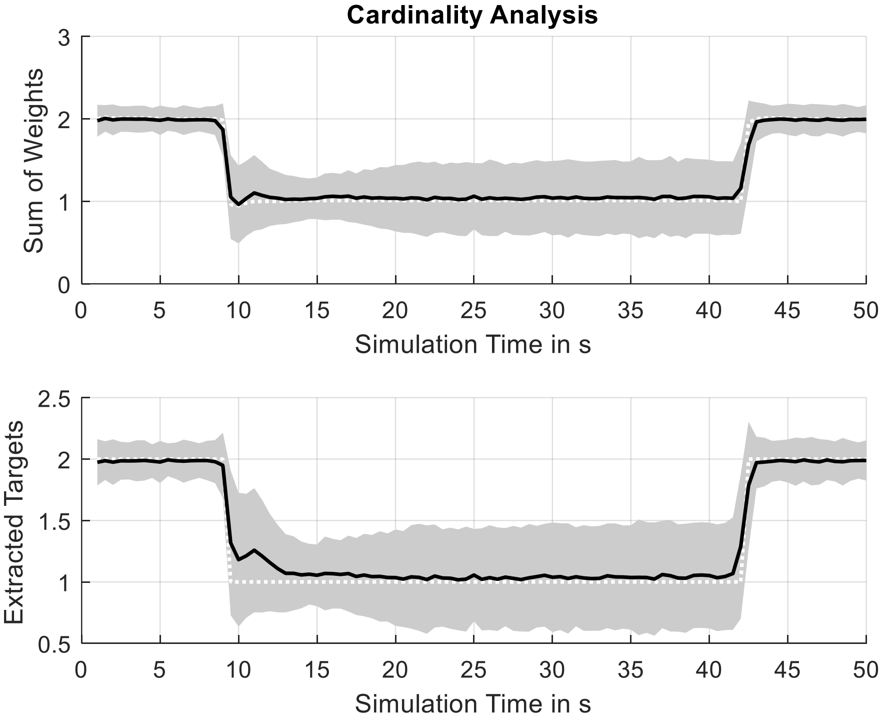

Figure 16: Cardinality evaluation of DP at 0.5m distance between objects with $\Sigma = 0.1m \cdot s^{-2}$. The black line is the mean value with the grey area showing the +/- standard deviation range. The white dotted line gives the median value.

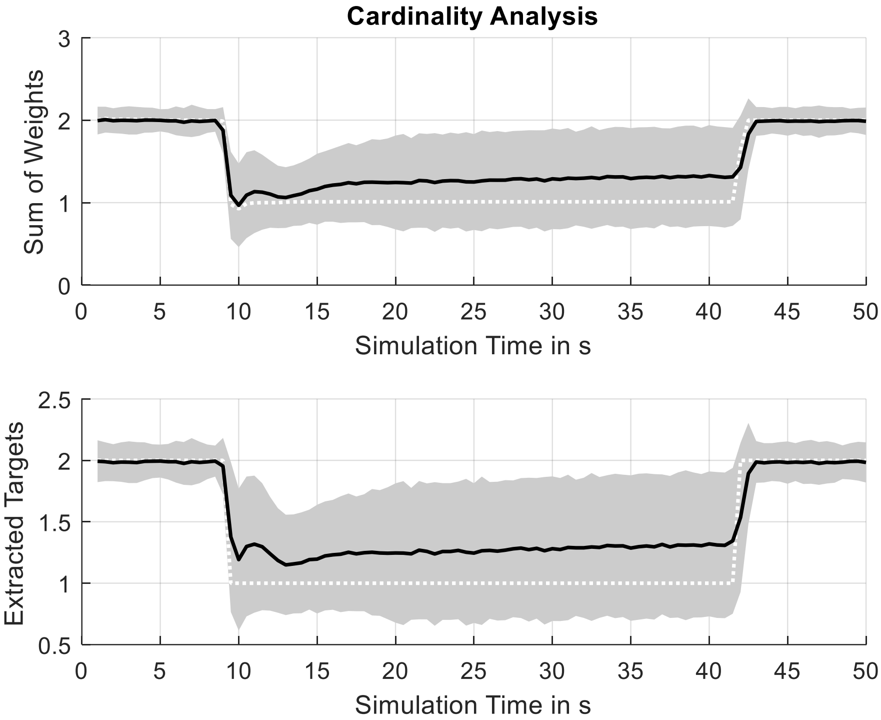

Figure 17: Cardinality evaluation of DP with SP at 0.5m distance between objects with $\Sigma = 0.1m \cdot s^{-2}$. The black line is the mean value with the grey area showing the +/- standard deviation range. The white dotted line gives the median value.

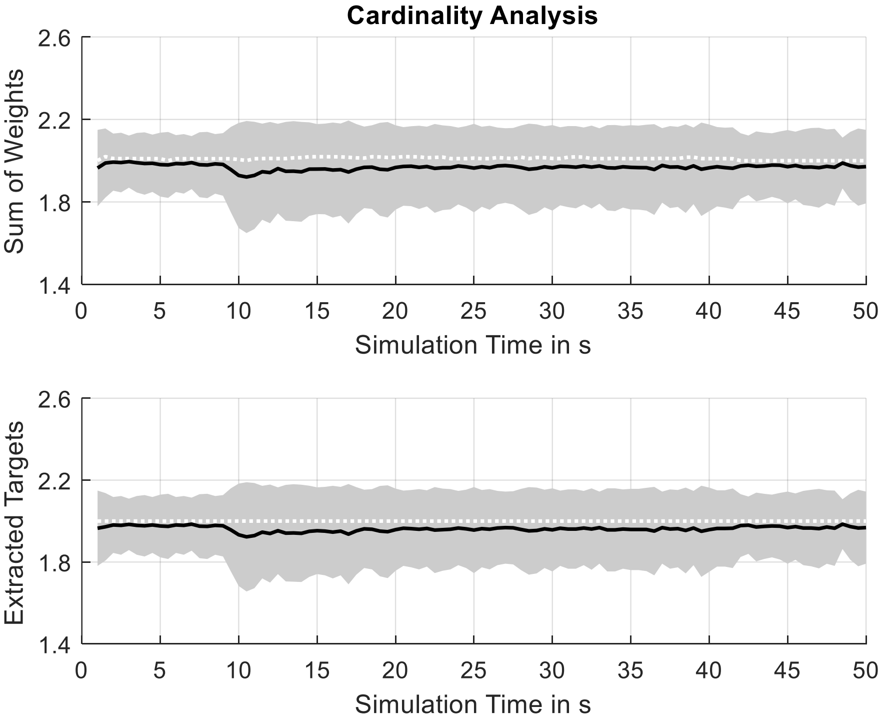

Figure 18: Cardinality evaluation of StP at 0.5m distance between objects with $\Sigma = 0.5m \cdot s^{-2}$. The black line is the mean value with the grey area showing the +/- standard deviation range. The white dotted line gives the median value.

In Figure 19, Figure 20 and Figure 21, the extent estimations of the particular partitioning methods are illustrated. The figures show the separation according to each major and minor half axis. The true values of the half-axes $A1, a1, A2$ and $a2$ were given in (8.6). Due to the fact that the filter using DP and SP underestimates the cardinality, it is implied that the objects are merged during the parallel motion period. Therefore, the extent estimation of the half-axes shows a bad performance. Comparing the half-axes estimates to StP, cf. Figure 21, it can be seen that the half-axes estimation converges to proper values which stays constant even when the objects move in parallel. This was already stated by the H-OSPA, which incorporates the extent estimation. At a simulation time of $42s$ the extent estimation for all half-axes becomes slightly worse because the objects are separating through driving a circle manoeuvre. In the filter equations, the heading prediction of the ellipse is not modelled explicitly. Due to this, for manoeuvring objects the extension estimation is in need of improvement as already mentioned in [6]. This can be done by incorporating the heading into the state vector and the estimation of its evolution over time.

The performance of DP and SP was shown to be worse than StP in this simulation setup. To improve this, it was spotted that the standard deviation of acceleration Σ has a big impact on the filter's performance using DP and SP. For the scenario with the distance of $0.5m$, additional MC simulations were done with the same Σ as for StP. Thus, with setting $\Sigma = 0.5\text{m} \cdot \text{s}^{-2}$, the process noise covariance $Q_{k|k-1}$ (4.14) increases, which influences the performance of the filter with DP and SP in a positive way. In Figure 22 and Figure 23 the cardinality estimates for DP and DP with SP are illustrated respectively, which show a much better performance compared to the setting with $\Sigma = 0.1\text{m} \cdot \text{s}^{-2}$. The filter is capable to separate the objects, whereas the additional use of SP, cf. Figure 23, leads the performance to become slightly better. Both methods tend to overestimate the cardinality in the phase, when the objects move in parallel. Compared to the cardinality estimation of StP shown in Figure 18, StP performs a tiny cardinality underestimation but with a less lower variance as DP and SP. Due to these insights, Σ was set to $0.5\text{m} \cdot \text{s}^{-2}$ for all partitioning methods in the scenario with an object distance of $0.0m$.

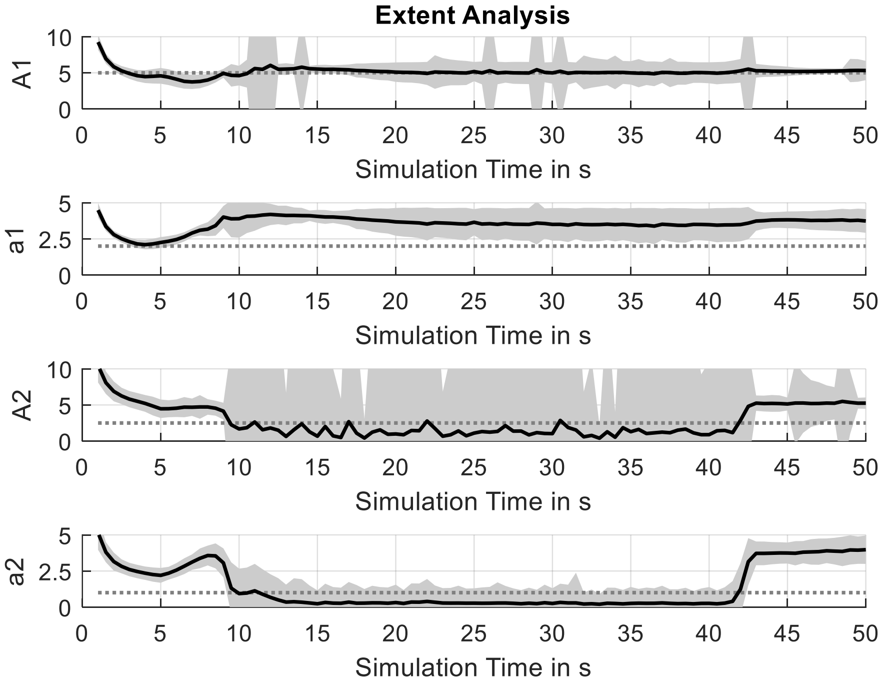

Figure 19: Half-axes estimation of DP at 0.5m distance between objects with $\Sigma = 0.1m \cdot s^{-2}$. The black line is the mean value with the grey area showing the +/- standard deviation range. The grey dotted line represents the true length of the half-axis in m.

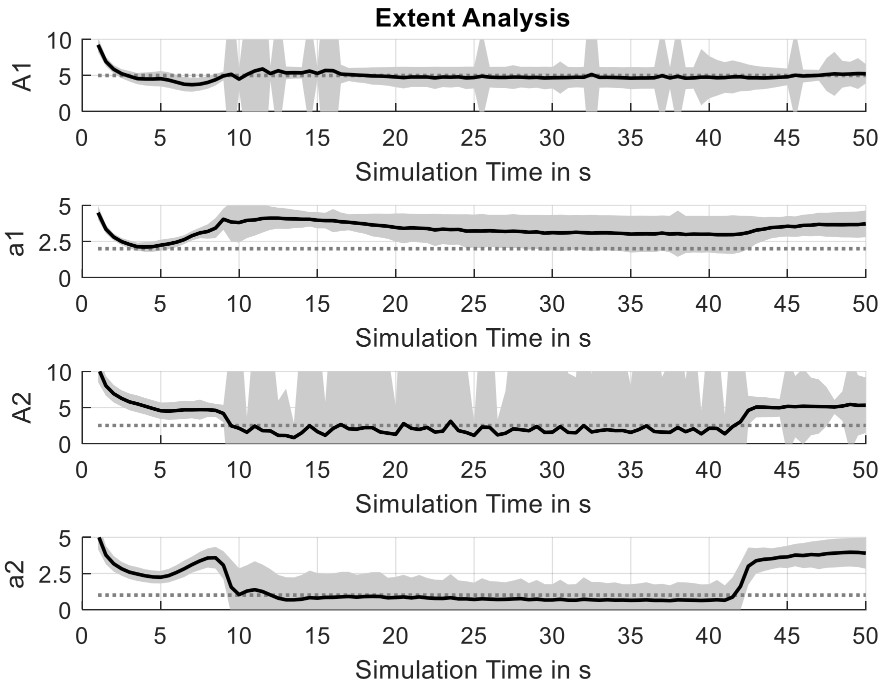

Figure 20: Half-axes estimation of DP with SP at 0.5m distance between objects with $\Sigma = 0.1m \cdot s^{-2}$*. The black line is the mean value with the grey area showing the +/- standard deviation range. The grey dotted line represents the true length of the half-axis in m.*

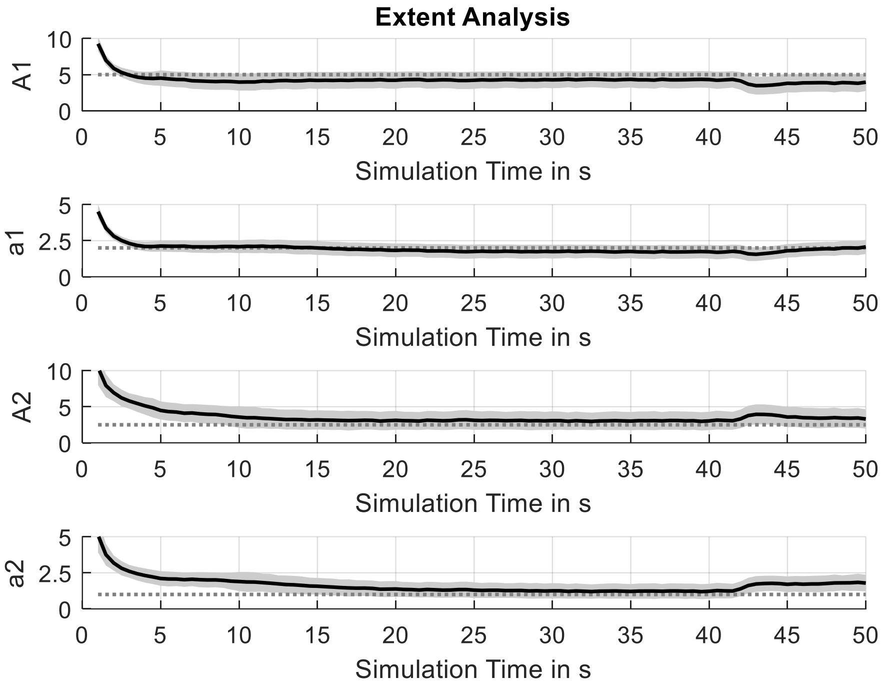

Figure 21: Half-axes estimation of DP with SP at 0.5m distance between objects with $\Sigma = 0.5m \cdot s^{-2}$*. The black line is the mean value with the grey area showing the +/- standard deviation range. The grey dotted line represents the true length of the half-axis in m.*

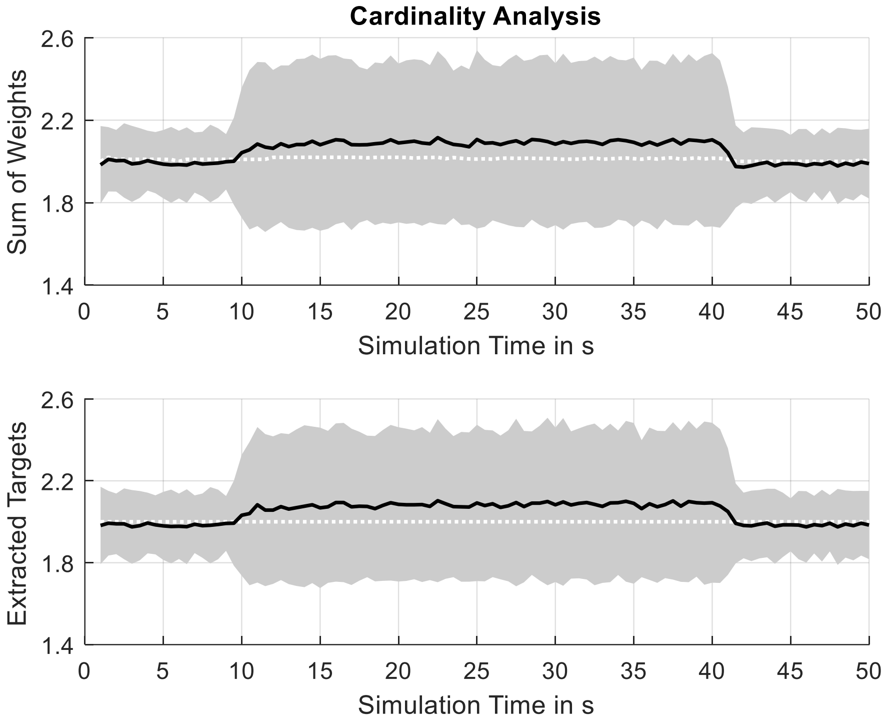

Figure 22: Cardinality evaluation of DP at 0.5m distance between objects with adapted $\Sigma = 0.5m \cdot s^{-2}$*. The black line is the mean value with the grey area showing the +/- standard deviation range. The white dotted line gives the median value.*

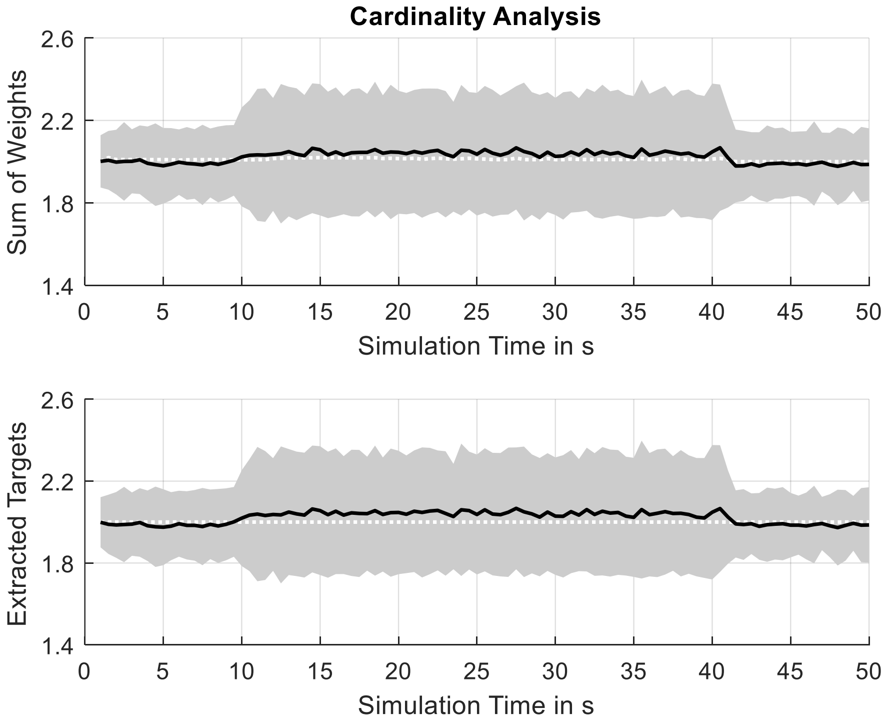

Figure 23: Cardinality evaluation of DP with SP at 0.5m distance between objects with adapted $\Sigma = 0.5m \cdot s^{-2}$*. The black line is the mean value with the grey area showing the +/- standard deviation range. The white dotted line gives the median value.*

8.4.2. Simulation results with a distance of $0.0m$

This subsection presents the MC simulation results for the parallel track scenario with a minimal object distance of $0.0m$. That means, the edges of the 3-sigma ellipse touch each other. As mentioned in the beginning of section 8.4, the results for the distance of $0.25m$ are given in the Appendix A.5. For the simulation results in this subsection, Σ was set to $0.5\mathrm{m} \cdot \mathrm{s}^{-2}$, to obtain the better performance of DP and SP as it was spotted through the simulations in subsection 8.4.1.

The OSPA distances for the use of DP are given in Figure 24. For the combination of DP with SP, see Figure 25 and for the use of StP, see Figure 26. The H-OSPA using DP shows a downwards trend until the objects come close to each other, what happens at a simulation time of $10s$. Afterwards, it arises to a value just above 0.4 until the objects separate. This is caused by the underestimation of the cardinality as it can be seen in Figure 27. At this close distance the filter performance suffers from the use of DP, which is not capable of separating the close objects. Further, this influences the OSPA, where a leap can be seen when the objects have approached.

With the additional use of SP, the performance of the filter is improved, cf. Figure 25. There, the H-OSPA reaches a plateau under a value of 0.3, when the objects are close. Further, the OSPA shows significantly lower values below a value of 10. When considering the corresponding cardinality estimation in Figure 28, it is revealed that the filter with the combination of DP and SP performs a more accurate cardinality estimation. The estimation is above a value of 2, and the filter tends to an overestimation. This behaviour is comparable with the ones at a distance of $0.25m$ and $0.5m$. The former is given in the Appendix A.5 and the latter was shown in Figure 23.

In Figure 26, the filter performance using StP is illustrated with the OSPAs. There, the H-OSPA further improves compared to DP and SP and converges to a level slightly below the value of 0.2 during the parallel motion phase. When the objects start separating at a simulation time of $40s$, the H-OSPA increases due to the fact of the turn motion. The better performance can be seen considering Figure 29, which shows the cardinality estimation using StP. The cardinality estimation process using StP is better compared to the filter using DP and SP. The estimate is close to the truth value of 2 with a tiny underestimation and a less variance of the cardinality.

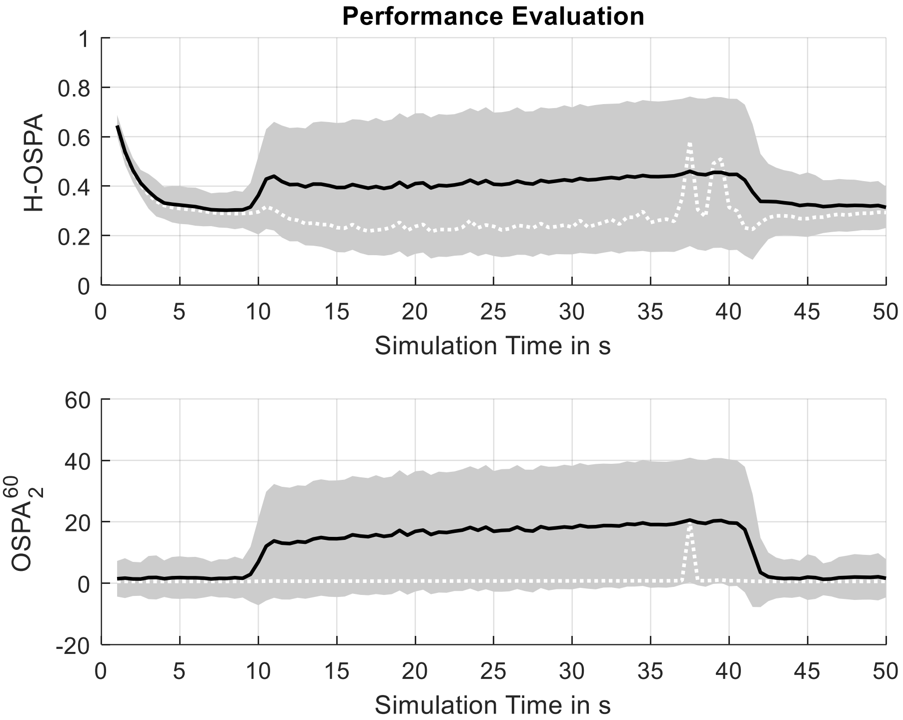

Figure 24: OSPA of DP at 0.0m distance between objects with $\Sigma = 0.5m \cdot s^{-2}$. The black line is the mean value with the grey area showing the +/- standard deviation range. The white dotted line gives the median value.

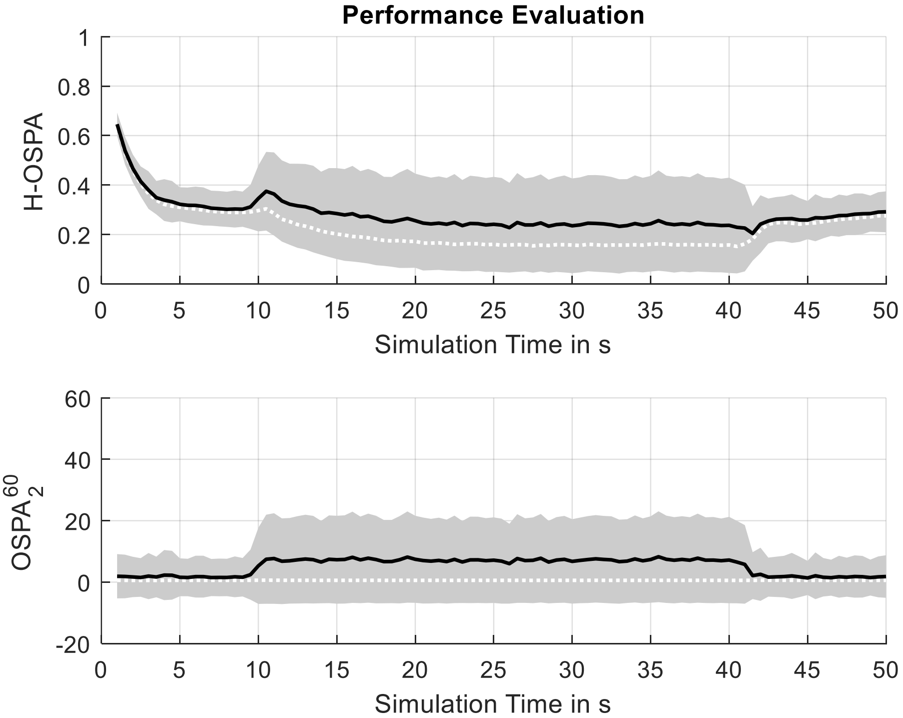

Figure 25: OSPA of DP with SP at 0.0m distance between objects with $\Sigma = 0.5m \cdot s^{-2}$. The black line is the mean value with the grey area showing the +/- standard deviation range. The white dotted line gives the median value.

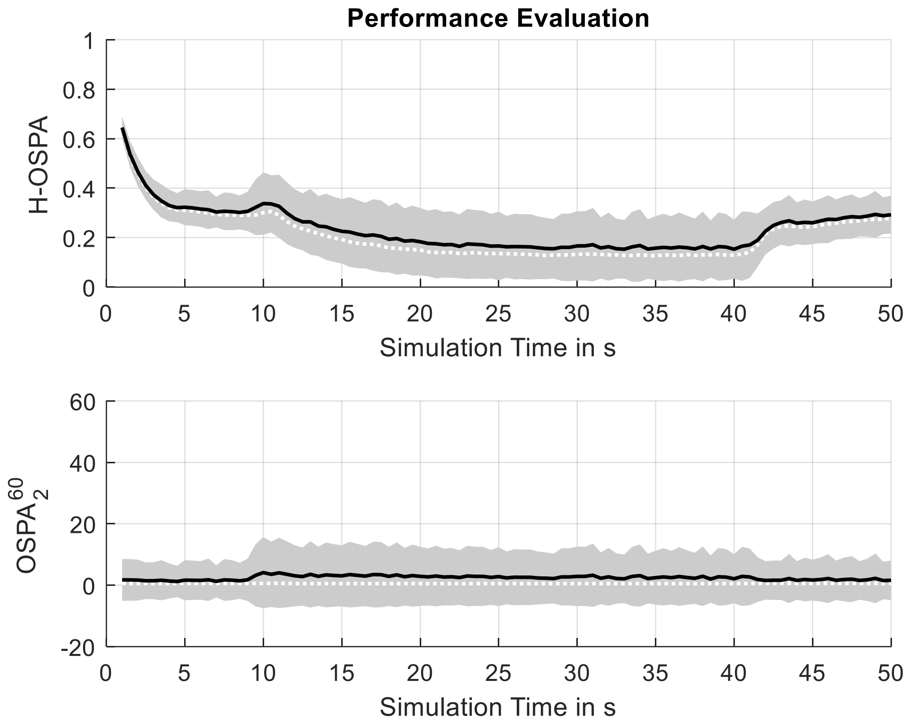

Figure 26: OSPA of StP at 0.0m distance between objects with $\Sigma = 0.5m \cdot s^{-2}$. The black line is the mean value with the grey area showing the +/- standard deviation range. The white dotted line gives the median value.

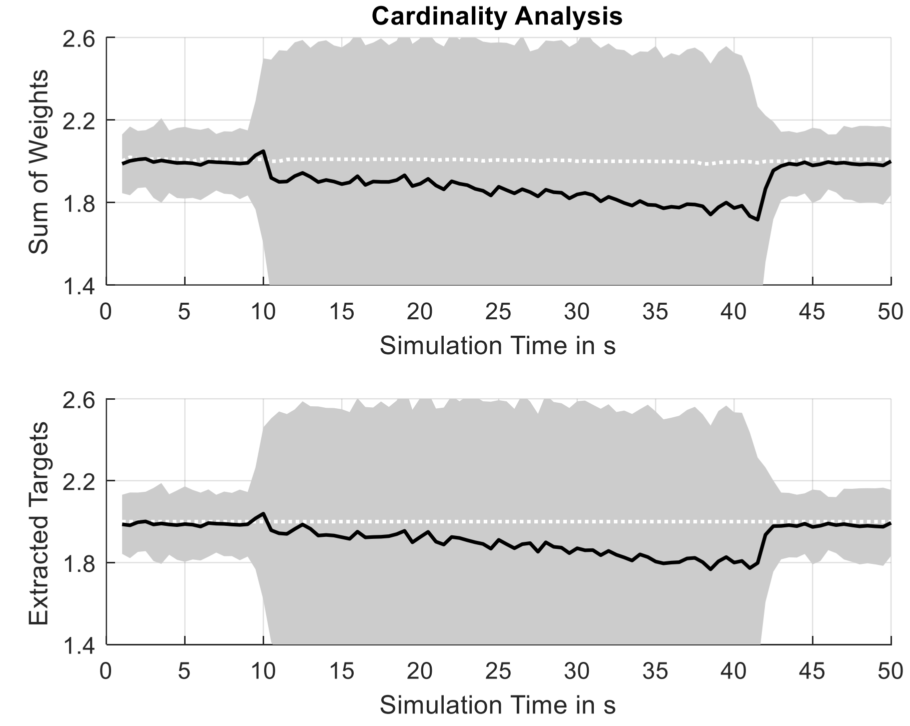

Figure 27: Cardinality evaluation of DP at 0.0m distance between objects with $\Sigma = 0.5m \cdot s^{-2}$. The black line is the mean value with the grey area showing the +/- standard deviation range. The white dotted line gives the median value.

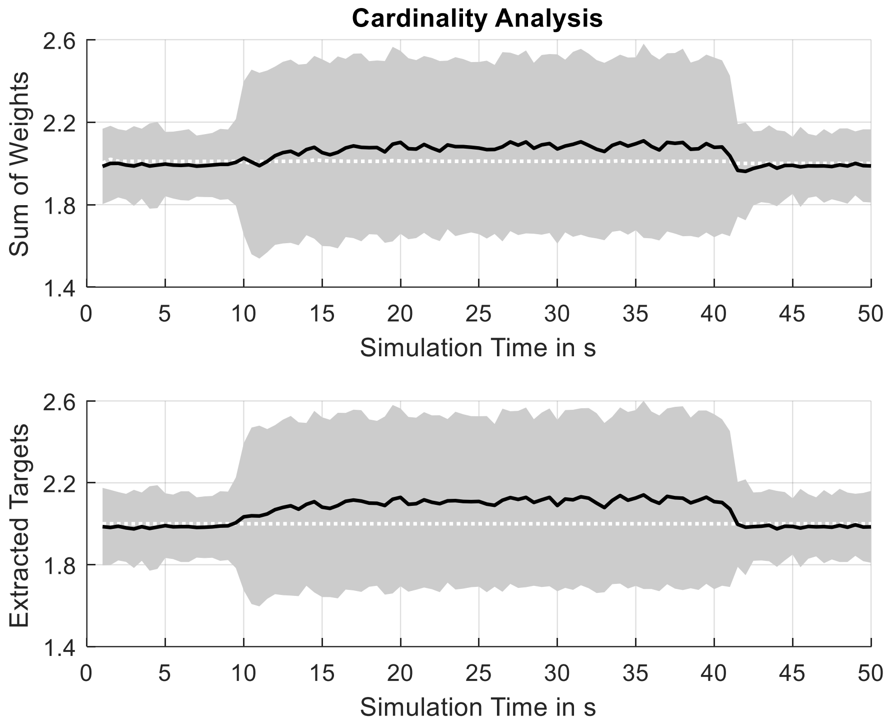

Figure 28: Cardinality evaluation of DP and SP at 0.0m distance between objects with $\Sigma = 0.5m \cdot s^{-2}$*. The black line is the mean value with the grey area showing the +/- standard deviation range. The white dotted line gives the median value.*

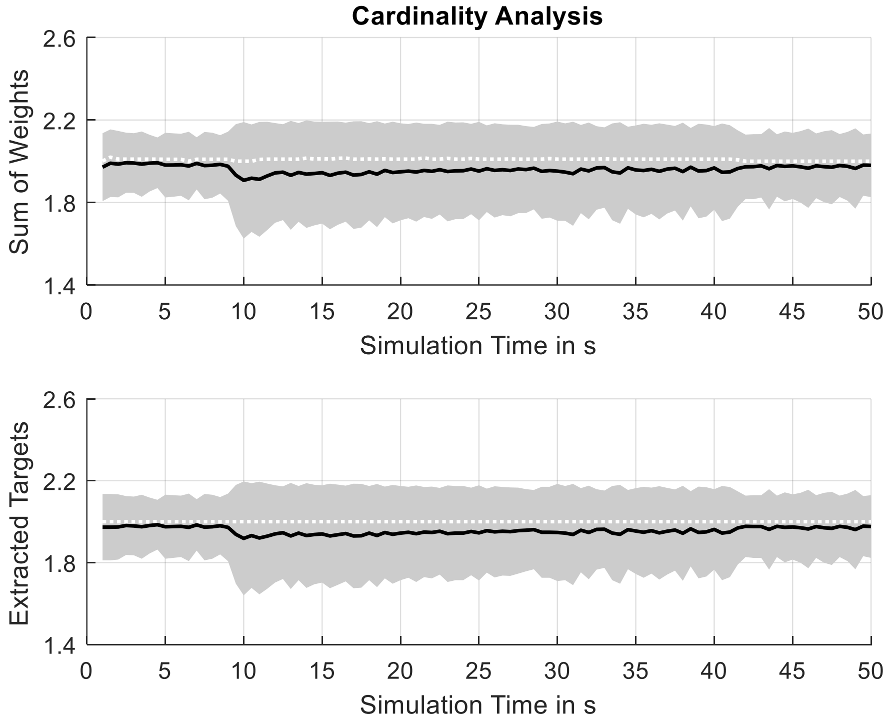

Figure 29: Cardinality evaluation of StP at 0.0m distance between objects with $\Sigma = 0.5m \cdot s^{-2}$*. The black line is the mean value with the grey area showing the +/- standard deviation range. The white dotted line gives the median value.*

Considering the extent analyses using DP, DP combined with SP and StP in Figure 30, Figure 31 and Figure 32, the better performance of the filter with StP is further confirmed. With the use of DP, the minor half-axes estimates $a1$ and $a2$ show constant high variances illustrated by the grey areas in Figure 30. Further, the major half-axes estimates have big additional outliers. With a regard on the results using DP with SP, cf. Figure 31, the performance is not improved significantly. The result of the MC run using StP is given in Figure 32, where it is clearly seen that this partitioning method delivers a stable output. There are no outliers present and the single axes estimates converge to a stable level while the objects move parallel. To demonstrate the better performance, the root mean squared errors (RMSEs) of the axes' estimates are given for all partitioning methods in Table 23.

The fact that the RMSE for $a1$ is smaller using DP than StP, should be considered with regard on the variances that are higher using DP, cf. Figure 30 and Figure 32.

Table 23: RMSEs of the half-axes estimation for all partitioning methods

	DP	DP/SP	StP
$A1$	$1.529m$	$1.627m$	$0.991m$
$a1$	$0.325m$	$0.534m$	$0.348m$
$A2$	$1.661m$	$1.946m$	$1.521m$
$a2$	$0.978m$	$1.001m$	$0.861m$

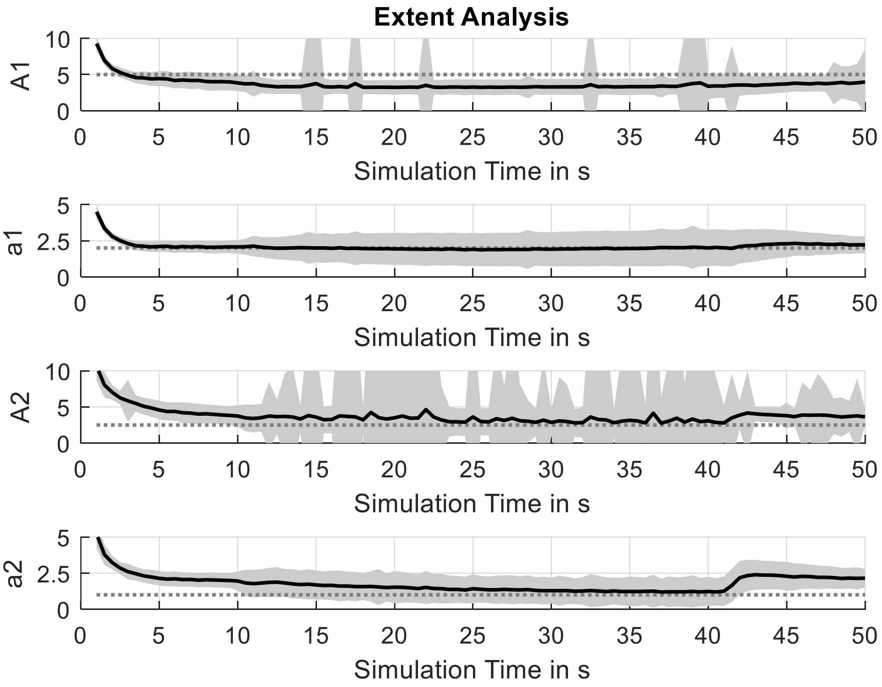

Figure 30: Half-axes estimation of DP at 0.0m distance between objects with $\Sigma = 0.5m \cdot s^{-2}$. The black line is the mean value with the grey area showing the +/- standard deviation range. The grey dotted line represents the true length of the half-axis in m.

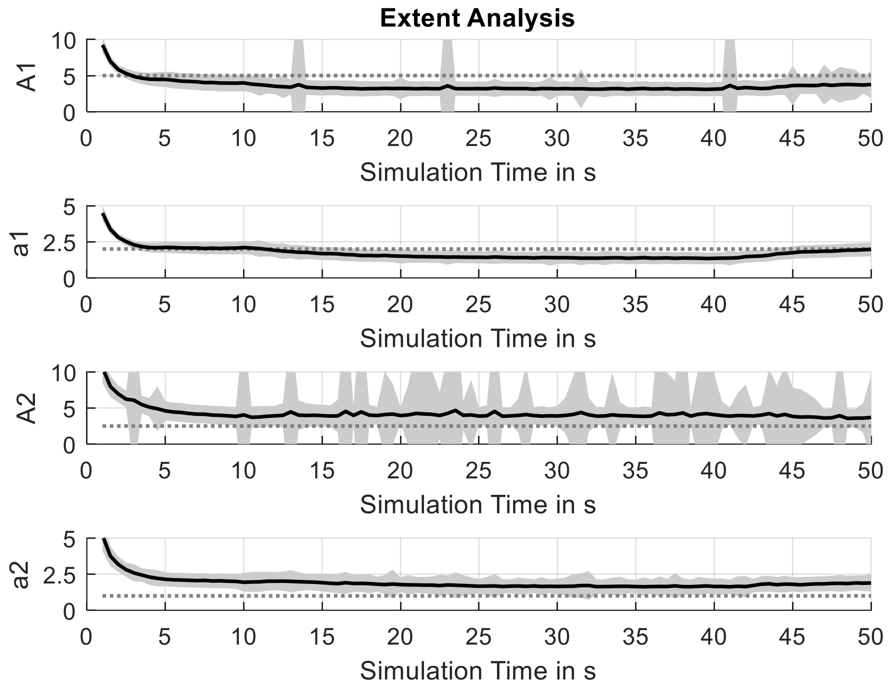

Figure 31: Half-axes estimation of DP with SP at 0.0m distance between objects with $\Sigma = 0.5m \cdot s^{-2}$. The black line is the mean value with the grey area showing the +/- standard deviation range. The grey dotted line represents the true length of the half-axis in m.

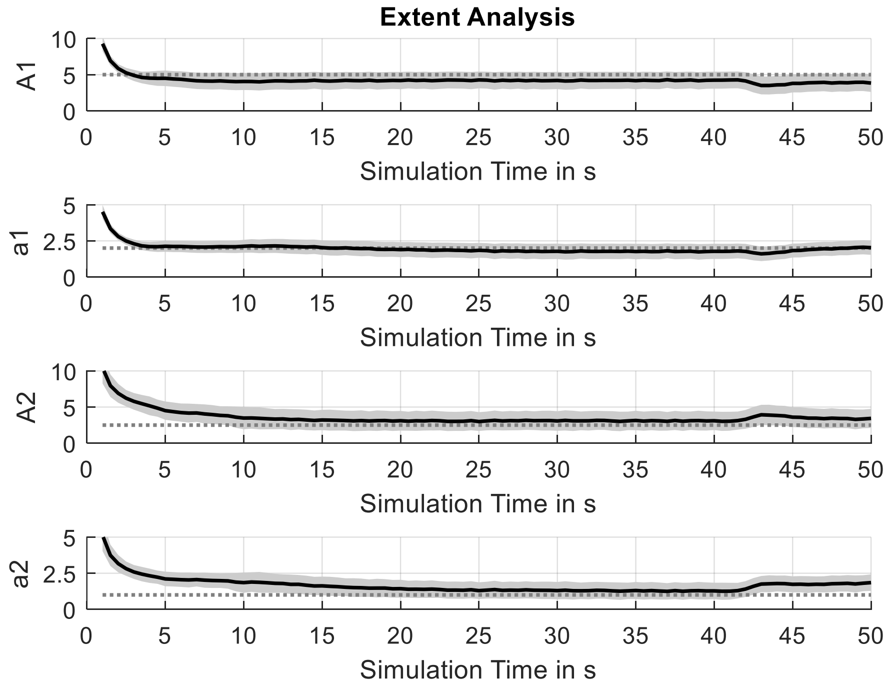

Figure 32: Half-axes estimation of StP at 0.0m distance between objects with $\Sigma = 0.5m \cdot s^{-2}$. The black line is the mean value with the grey area showing the +/- standard deviation range. The grey dotted line represents the true length of the half-axis in m.

As a last evaluation criterion for the three partitioning methods, the computational cost is considered. Therefore, the averaged execution times are illustrated in Figure 33, Figure 34 and Figure 35.

The averaged execution times were calculated as follows. The execution time is cumulated separately for each single filter step over one simulation run. Then the averaged execution time, as illustrated in the figures, is calculated as the mean over all 1000 MC runs.

The *Target Extraction, Pruning/Merging* and P*rediction* step are illustrated in the figures for the sake of completeness. However, the execution time of the *Partitioning, Filter Correction* and *Correction Components* step are of major interest, because they are mainly influenced using the different partitioning methods. Note that the step *Correction Components* considers the calculation of the components measurement centroid $\bar{\mathbf{z}}_k$ and scatter matrix $\bar{Z}_k$ defined below (4.38). Those are separated from the main *Filter Correction* step as already done in the original implementation [43, p. 2]. The *Partitioning* step captures the time that is required for the partitioning using DP or DP combined with SP. Equivalently, the *Stochastic Partitioning* step captures the time for the StP actions over all sampling iterations t. This contains the time demanded by the calculations given in Table 14, but without the correction parts ($correction\_step(\cdot)$, $partial\_correction\_step(\cdot)$). These are captured in the execution time values of *Filter Correction*.

Figure 33 and Figure 34 show the results for DP and DP combined with SP. It can be seen for DP, that the partitioning step needs the highest computational effort with roughly $1.5s$ per simulation run. Followed by the *Filter Correction* step that is more than ten times lower with it's maximum of roughly $0.1s$ per run. The step *Correction Components* is negligible small with about $3.5ms$. The *Partitioning* step's computational effort stays constant over the simulation time, which is caused by the fix number of partitions that have to be computed every filter time step k. The effort for the *Filter Correction* increases during the parallel motion phase of the objects. When comparing these results to the case when DP is used in combination with SP, cf. Figure 34, the *Partitioning* step needs nearly the same effort. The differences can not be distinguished with these figures. Therefore, one additional depiction is given in the Appendix A.6. There it can be seen, that the maximum additional execution time of SP is roughly $60ms$ compared to the effort of about $1.5s$ of DP. The plateau of $60ms$ mirrors in the *Filter Correction* and *Correction Components* step calculation of Figure 34.

In the execution time analysis of StP, cf. Figure 35. the main computational effort is generated by the *Filter Correction* step with about $4s$ per simulation run. This is about 40 times higher compared to the *Filter Correction* step using DP and SP. The *Stochastic Partitioning* step is with roughly $1.3s$ per run within a similar magnitude than the *Partitioning* step DP. Further, the effort for calculating the *Correction Components* is increased to a tenfold. These increases are explainable with the higher number of corrections that are executed during the StP step, cf. Table 14. The correction step is executed in advance of the StP action for an initial partition denoted as $correction\_step(\cdot)$. During the iterations, where every iteration changes one or two cells, the resulting partitions are used to do further correction steps, denoted as $partial\_correction\_step(\cdot)$. This means, only the components that are affected by the StP actions are recalculated, cf. line 9 in Table 14, while all other components are fixed. Now one can imagine, that without the $partial\_correction\_step(\cdot)$, the computational cost would burst. All cells of a changed partition $\mathscr{p}_{\alpha}^{(t)}$ would be corrected every sampling iteration t despite there are cells that already have been corrected doing the initial correction step, cf. line 3 in Table 14.

To summarise the consideration of the computational cost for all three methods, it can be stated, that the better performance of StP is obtained with the over three times higher computational effort compared to DP and SP. This statement is valid for the simulated setup and could vary for different scenarios. Further, the DP method was setup to deliver the best tracking performance, not the lowest computational cost. The StP algorithm could be optimised in a way that the number of iterations could be decreased, because it was emphasized that the algorithm converges after roughly a fourth of its iterations. That means, the iteration factor T_{StP}, cf. Table 14, could be chosen with a lower value. All these optimisations are only possible with the a priori knowledge of the scenarios and in unknown scenarios this would be not possible.

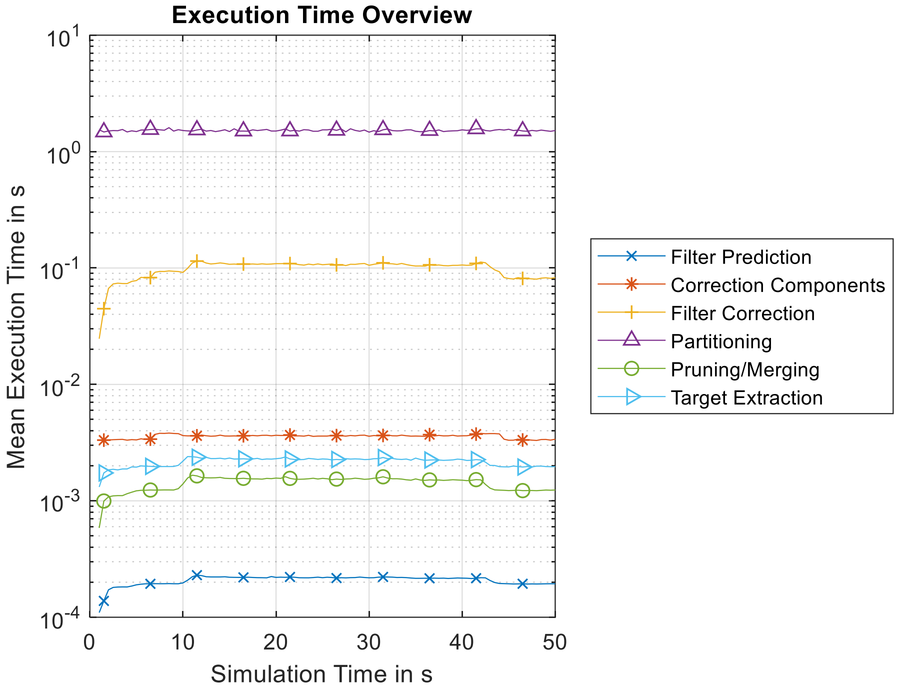

Figure 33: Averaged execution time of the single filter steps of DP at 0.0m distance between objects with $\Sigma = 0.5m \cdot s^{-2}$.

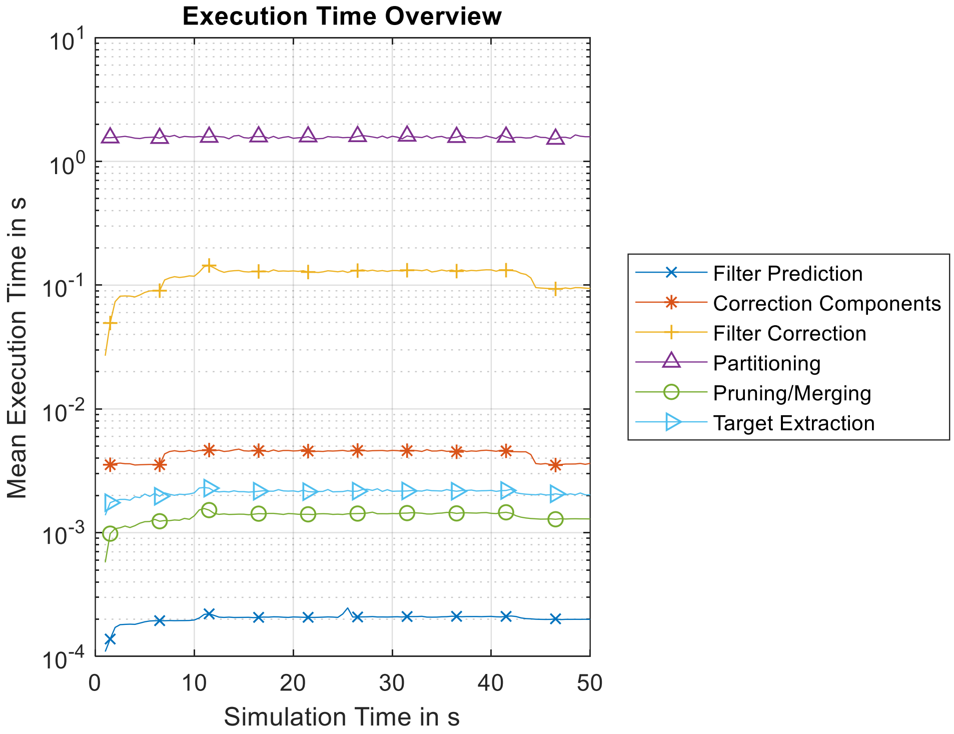

Figure 34: Averaged execution time of the single filter steps of DP with SP at 0.0m distance between objects with $\Sigma = 0.5m \cdot s^{-2}$.

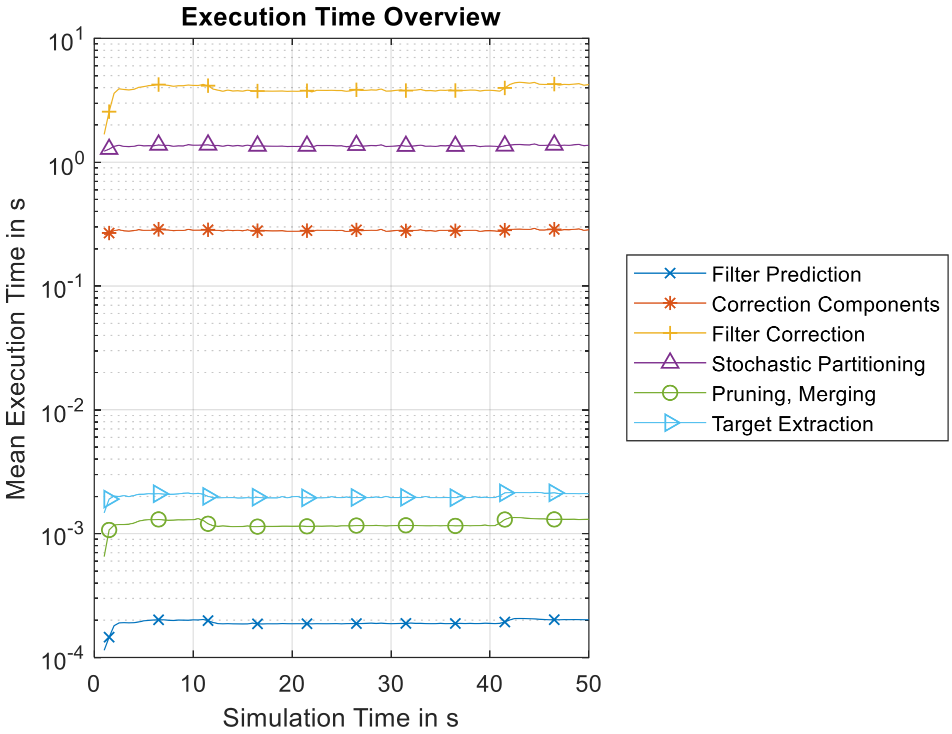

Figure 35: Averaged execution time of the single filter steps of StP at 0.0m distance between objects with $\Sigma = 0.5m \cdot s^{-2}$*.*

9. Conclusion

Within this thesis, the fundamentals of object tracking with its varieties including the state estimation problem, the multi-object tracking problem, the extended object tracking problem and the multiple extended object tracking problem were presented in a detailed way. Further, it was differentiated between classical data association methods and likelihood-based methods. The approaches Distance Partitioning, Sub-Partitioning as well as the method of Granström, called Stochastic Optimisation, were provided and discussed in detail. Based on these considerations, the likelihood-based partitioning method Stochastic Partitioning was developed and proposed for the GIW-PHD filter. The implementation of this new partitioning method was provided and it was shown that it was successfully integrated into the filter framework. Further, the GIW-PHD filter with its classical partitioning methods was implemented to allow a scientific comparison between the filters. The comparison was done using a simulation environment that creates random scenarios as well as a defined parallel tracking scenario. The latter was evaluated using Monte Carlo simulations. The random scenarios have shown that the filters can handle object actions like birth and death. The spawning of objects was revealed to work for classical heuristic methods in a better way than for Stochastic Partitioning. To handle spawning in a really reliable way, it should be explicitly modelled within the filter equations using e.g. [61]. The Monte Carlo simulation results of the parallel track scenarios have shown, that the filter implementation using Stochastic Partitioning shows better performance. This is emphasized with lower miss distance values of the OSPA and the H-OSPA as well as with the more accurate extent estimation. Further, for all different distances between the objects, the filter using Stochastic Partitioning delivers a very stable performance with low variances of the single runs. The runtime analysis has shown that the computational cost using Stochastic Partitioning has increased over three times for the defined parallel tracking scenario.

Further work could be done by an investigation of the proposed implementation using real world measurement data of similar scenarios. This creates the need to incorporate nonlinear models into the filter equations, because in general sensor measurements are collected with range and azimuth. Another investigation could be the incorporation of a heading prediction for the elliptical objects, which would further improve the performance of the likelihood-based partitioning and thus the overall filter performance for manoeuvring objects. Of further interest for the Institute of System Dynamics would be a direct performance comparison of the proposed filter with the developed multi-detection JIPDA filter of [2]. Consequently, there arise a lot of interesting and perspective investigations to be researched on in the future.

II. List of abbreviations

bmf	belief mass function
CPHD	Cardinalized Probability Hypothesis Density
DP	Distance Partitioning
EAP	expectation a posteriori
EKF	Extended Kalman filer
EOS	Extended object state
FISST	Finite Set Statistics
GIW	Gaussian inverse Wishart
GM	Gaussian mixture
GNN	Global nearest neighbour
HOSPA	Hellinger distance OSPA
i.i.d.	Independent identically distributed
JIPDA	Joint Integrated PDA
LKF	Linear Kalman filter
MAP	maximum a posteriori
MC	Monte Carlos
MD	Multi detection
MH	Metropolis Hastings
MHT	Multiple hypothesis tracking
MOT	Multi-object tracking
NN	Nearest neighbour
OSPA	Optimal sub-pattern assignment
p.g.fl.	probability generating functionals
PDA	Probabilistic data association
pdf	Probability density function
PHD	Probability hypothesis density
PMBM	Poisson multi-Bernoulli mixture
pmf	probability mass function
PPP	Poisson point process
RFS	Random finite set
RFS	random finite set
SECRS	simulation environment creating random scenarios
SO	Stochastic Optimisation
SP	Sub-Partitioning
SPD	square positive definite
StP	Stochastic Partitioning
UKF	unscented Kalman filter

III. List of figures

IV. List of tables

V. Appendix

A.1. GM-PHD filter pseudo code

Here, the pseudo code for the Gaussian Mixture PHD filter implementation of Vo et al. [12] is depicted. Explanations of notation can be looked up in section 3.5, where the filter is introduced.

Table 24: The GM-PHD filter algorithm

Given Gaussian mixture components $\left\{w_{k-1}^{(i)}, \boldsymbol{m}_{k-1}^{(i)}, P_{k-1}^{(i)}\right\}_{i=1}^{J_{k-1}}$ at time step k, measurement set $\boldsymbol{Z}_k$ at time step k

Step 1 – Prediction of birth and spawn objects,
Birth objects:

```
$i = 0$
for $j = 1, \dots, J_{\gamma,k}$
    $i = i + 1$
    $w_{k|k-1}^{(i)} = w_{\gamma,k}^{(j)},\ \boldsymbol{m}_{k|k-1}^{(i)} = \boldsymbol{m}_{\gamma,k}^{(j)},\ P_{k|k-1}^{(i)} = P_{\gamma,k}^{(j)}$
end for
```

Spawn objects:

```
for $j = 1, \dots, J_{\beta,k}$
    for $l = 1, \dots, J_{k-1}$
        $i = i + 1$
        $w_{k|k-1}^{(i)} = w_{k-1}^{(l)} w_{\beta,k}^{(j)}$
        $\boldsymbol{m}_{k|k-1}^{(i)} = \boldsymbol{d}_{\beta,k-1}^{(j)}$[21] $+ F_{\beta,k-1}^{(j)} \boldsymbol{m}_{k-1}^{(l)}$
        $P_{k|k-1}^{(i)} = Q_{\beta,k-1}^{(j)} + F_{\beta,k-1}^{(j)} P_{k-1}^{(l)} \left(F_{\beta,k-1}^{(j)}\right)^T$
    end for
end for
```

Step 2 – Prediction for existing objects:

```
for $j = 1, \dots, J_{k-1}$
    $i = i + 1$
    $w_{k|k-1}^{(i)} = p_{S,k} w_{k-1}^{(j)}$
    $\boldsymbol{m}_{k|k-1}^{(i)} = F_{k-1} \boldsymbol{m}_{k-1}^{(j)}$
    $P_{k|k-1}^{(i)} = Q_{k-1} + F_{k-1} P_{k-1}^{(j)} (F_{k-1})^T$
end for
$J_{k|k-1} = i$
```

Step 3 – Construction of PHD update components:

```
for $j = 1, \dots, J_{k|k-1}$
    $\eta_{k|k-1}^{(j)} = H_k \boldsymbol{m}_{k|k-1}^{(j)}, S$
    $S_k^{(j)} = R_k + H_k P_{k|k-1}^{(j)} (H_k)^T$
    $K_k^{(j)} = P_{k|k-1}^{(j)} (H_k)^T \left(S_k^{(j)}\right)^{-1}$
    $P_{k|k}^{(j)} = \left[I_d - K_k^{(j)} H_k\right] P_{k|k-1}^{(j)}$
end for
```

[21] This is a kind of distance offset, where the spawning component is modelled to appear around the predicted mean.

Step 4 – Update of PHD components,

```
Not detected objects:
for $j = 1, \dots, J_{k|k-1}$
    $w_k^{(j)} = \left(1 - p_{D,k}\right) w_{k|k-1}^{(j)}$
    $\boldsymbol{m}_k^{(j)} = \boldsymbol{m}_{k|k-1}^{(j)}, P_k^{(j)} = P_{k|k-1}^{(j)}$
end for
Detected objects:
$l = 0$
for each $\boldsymbol{z} \in \boldsymbol{Z}_k$
    $l = l + 1$
    for $j = 1, \dots, J_{k|k-1}$
        $w_k^{(l \cdot J_{k|k-1} + j)} = p_{D,k} w_{k|k-1}^{(j)} \mathcal{N}\left(\boldsymbol{z}; \eta_{k|k-1}^{(j)}, S_k^{(j)}\right)$
        $\boldsymbol{m}_k^{(l \cdot J_{k|k-1} + j)} = \boldsymbol{m}_{k|k-1}^{(j)} + K_k^{(j)}\left(\boldsymbol{z} - \eta_{k|k-1}^{(j)}\right)$
        $P_k^{(l \cdot J_{k|k-1} + j)} = P_{k|k}^{(j)}$
    end for
    $w_k^{(l \cdot J_{k|k-1} + j)} = \frac{w_k^{(l \cdot J_{k|k-1} + j)}}{\kappa_k(\boldsymbol{z}) + \sum_{i=1}^{J_{k|k-1}} w_k^{(l \cdot J_{k|k-1} + i)}}$ for $j = 1, \dots, J_{k|k-1}$
end for
$J_k = l \cdot J_{k|k-1} + J_{k|k-1}$
```

Output updated Gaussian mixture components $\left\{w_k^{(i)}, \boldsymbol{m}_k^{(i)}, P_k^{(i)}\right\}_{i=1}^{J_k}$ at time step k

In Table 25, the pseudo code for pruning and merging of the filter components is given [12]. This scheme is used to stem the expeditiously increase of components after step 4 in Table 24.

Table 25: Pruning and merging algorithm for the GM-PHD filter

Given updated Gaussian mixture components $\left\{w_k^{(i)}, \boldsymbol{m}_k^{(i)}, P_k^{(i)}\right\}_{i=1}^{J_k}$ at time step k, a truncation threshold T, a merging threshold U, and a maximum allowable number of GM components J_{max}[22]

```
set $l = 0$ and $\boldsymbol{I} = \{i = 1, \dots, J_k | w_k^{(i)} > T\}$
repeat
    $l = l + 1$
    $j = \arg\max_{i \in \boldsymbol{I}} w_k^{(i)}$
    $\boldsymbol{L} = \left\{ i \in \boldsymbol{I} \middle| \left(\boldsymbol{m}_k^{(i)} - \boldsymbol{m}_k^{(j)}\right)^T \left(P_k^{(i)}\right)^{-1} \left(\boldsymbol{m}_k^{(i)} - \boldsymbol{m}_k^{(j)}\right) \leq U \right\}$
    $\widetilde{w}_k^{(l)} = \sum_{i \in \boldsymbol{L}} w_k^{(i)}$
    $\widetilde{\boldsymbol{m}}_k^{(l)} = \frac{1}{\widetilde{w}_k^{(l)}} \sum_{i \in \boldsymbol{L}} w_k^{(i)} \boldsymbol{m}_k^{(i)}$
    $\widetilde{P}_k^{(l)} = \frac{1}{\widetilde{w}_k^{(l)}} \sum_{i \in \boldsymbol{L}} w_k^{(i)} \left(P_k^{(i)} + \left(\widetilde{\boldsymbol{m}}_k^{(l)} - \boldsymbol{m}_k^{(i)}\right)\left(\widetilde{\boldsymbol{m}}_k^{(l)} - \boldsymbol{m}_k^{(i)}\right)^T\right)$
    $\boldsymbol{I} = \boldsymbol{I} \backslash \boldsymbol{L}$
until $\boldsymbol{I} = \emptyset$
if $l > J_{max}$ then replace $\left\{\widetilde{w}_k^{(i)}, \widetilde{\boldsymbol{m}}_k^{(i)}, \widetilde{P}_k^{(i)}\right\}_{i=1}^{l}$ by those of the $J_{max}$ Gaussian mixture components with largest weights
```

Output pruned and merged Gaussian mixture components $\left\{\widetilde{w}_k^{(i)}, \widetilde{\boldsymbol{m}}_k^{(i)}, \widetilde{P}_k^{(i)}\right\}_{i=1}^{l}$ at time step k

In Table 26, the pseudo code for the multi-object state extraction is given that is used to create a simple state estimation of the objects $\widehat{\boldsymbol{X}}_k$ [12].

Table 26: Multi-object state extraction for the GM-PHD filter

Given Gaussian mixture components $\left\{w_k^{(i)}, \boldsymbol{m}_k^{(i)}, P_k^{(i)}\right\}_{i=1}^{J_k}$ at time step k

```
set $\widehat{\boldsymbol{X}}_k = \emptyset$
for $i = 1, \dots, J_k$
    if $w_k^{(i)} > 0.5$
        for $j = 1, \dots,$ round $(w_k^{(i)})$
            update $\widehat{\boldsymbol{X}}_k \triangleq \left[\widehat{\boldsymbol{X}}_k, \boldsymbol{m}_k^{(i)}\right]$
        end for
    end if
end for
```

Output multi-object state estimate $\widehat{\boldsymbol{X}}_k$ with Gaussian mixture components

[22] In Vo et al. [12] parameters are set to $T = 10^{-5}$, $U = 4$ and $J_{max} = 100$.

A.2. The Kronecker product and Kronecker delta

The Kronecker product and Kronecker delta are named after the German mathematician Leopold Kronecker and are defined as follows.

1. Kronecker product

The Kronecker product is denoted as $\otimes$ and defines a multiplication of two matrices of arbitrary shape.
For two matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{o \times p}$, with

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix},$$

the Kronecker product is defined as

$$C = A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix}.$$

The resulting matrix C is of dimension $\mathbb{R}^{(m \cdot o) \times (n \cdot p)}$, i.e. has $m \cdot o$ rows and $n \cdot p$ columns.

2. Kronecker delta

The Kronecker delta denoted as $\delta_{i,j}$ is a mathematical symbol that operates on an index set $\boldsymbol{S}$. The two elements i, j are included in $\boldsymbol{S}$, i.e. $i, j \in \boldsymbol{S}$. Thus the Kronecker delta is defined as

$$\delta_{i,j} = \begin{cases} 1 & if\ i = j \\ 0 & if\ i \neq j \end{cases}.$$

A.3. Probability densities

1. Multivariate Gaussian density

If a random vector $\boldsymbol{x} \in \mathbb{R}^d$ is Gaussian distributed with mean $\boldsymbol{m} \in \mathbb{R}^d$ and covariance $\Sigma \in \mathbb{R}^{dxd}$ it is noted as

$$\boldsymbol{x} \sim \mathcal{N}_d(\boldsymbol{m}, \Sigma).$$

Thus the Gaussian probability density function is

$$\mathcal{N}(\boldsymbol{x}; \boldsymbol{m}, \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m})^{\mathrm{T}} \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{m})},$$

with expectation

$$\mathbb{E}\{\boldsymbol{x}\} = \boldsymbol{m}$$

and covariance matrix

$$\mathbb{C}\{\boldsymbol{x}\} = \Sigma.$$

2. Wishart density

A SPD random matrix $X \in \mathbb{R}^{d \times d}$ is Wishart distributed if the probability density function is given as [2]

$$\mathcal{W}_d(X; n, \Sigma) = \frac{\sqrt{|X|^{n-d-1}}}{\boldsymbol{\Gamma}_d\left(\frac{n}{2}\right)\sqrt{2^{dn}|\Sigma|^n}}, e^{-\frac{1}{2}tr(\Sigma^{-1}X)}, n \geq d,$$

where X is the product SS^T containing the matrix $S = [\boldsymbol{s}^{(1)}, \dots, \boldsymbol{s}^{(n)}]$ that consists of n independent normal distributed vectors $\boldsymbol{s}^{(i)} \sim \mathcal{N}_d(\mathbf{0}, \Sigma)$ with $\boldsymbol{s}^{(i)} \in \mathbb{R}^d$. Further, it must hold that the degrees of freedom $n > d - 1$ and the scale matrix $\Sigma > 0$. $tr(\cdot)$ is the trace function and $\boldsymbol{\Gamma}_d(\cdot)$ is the multivariate Gamma function defined as

$$\boldsymbol{\Gamma}_d(a) = \pi^{\frac{d^2-d}{4}} \prod_{i=1}^{d} \Gamma\left(a + \frac{1-i}{2}\right),$$

with the gamma function $\Gamma(b) = (b-1)!$
The expectation SPD matrix is

$$\mathbb{E}\{X\} = n\Sigma$$

and the covariance matrix is

$$\mathbb{C}\{X\} = \mathbb{E}\{XX\} - \mathbb{E}\{X\}^2 = n(\Sigma\Sigma + tr(\Sigma)\Sigma)$$

A random matrix X that is Wishart distributed with the properties above is denoted as $X \sim \mathcal{W}_d(n, \Sigma)$.

3. *Inverse Wishart density*

If the random SPD matrix $Y \in \mathbb{R}^{d \times d}$ is inverse Wishart distributed, then the probability density function is [2]

$$\mathcal{IW}_d(Y; m, \Phi) = \frac{\sqrt{|\Phi|^{m-d-1}}}{\boldsymbol{\Gamma}_d\left(\frac{m-d-1}{2}\right)\sqrt{2^{m-d-1}|Y|^m}} e^{-\frac{1}{2}tr(Y^{-1}\Phi)},$$

where in both cases the trace function $tr(\cdot)$ and the multivariate Gamma function $\boldsymbol{\Gamma}_d$ are defined in the Appendix A.3., section 2. m is the degree of freedom and Φ the inverse scale matrix. The inverse Wishart distributed matrix is denoted as $Y \sim \mathcal{IW}_d(m, \Phi)$. The first and second order moments are given respectively as

$$\mathbb{E}\{Y\} = \frac{1}{m - 2d - 2}$$

and

$$\mathbb{C}\{Y\} = \frac{(m-2d-2)tr(\Phi)\Phi + (m-2d)\Phi\Phi}{(m-2d-1)(m-2d-2)^2(m-2d-4)}.$$

The inverse Wishart distribution and Wishart distribution reveal a relationship. If $Y \sim \mathcal{IW}_d(m, \Phi)$ then the relation to the Wishart distribution is

$$Y^{-1} \sim \mathcal{W}(m - d - 1, \Phi^{-1}).$$

Note that the inverse Wishart density can be defined in an alternative way [40] given as

$$\mathcal{IW}_d(Y; m, \Phi) = \frac{\sqrt{|\Phi|^m}}{\boldsymbol{\Gamma}_d\left(\frac{m}{2}\right)\sqrt{2^{md}|Y|^{m+d+1}}} e^{-\frac{1}{2}tr(\Phi Y^{-1})}.$$

For the definition of first and second moments, which change due to the different definition, see [38].

A.4. Pruning and merging for the GIW-PHD filter

In Table 27, the pseudo code for pruning and merging of the Gaussian inverse Wishart components is given [43]. This scheme is used to stem the rapid increase of components after prediction and correction in subsection 5.3.4.

Table 27: Pruning and merging algorithm for the GIW-PHD filter

Given corrected GIW mixture components $\left\{w_{k|k}^{(n)}, \xi_{k|k}^{(n)}\right\}_{n=1}^{J_{k|k}}$ at time step k, a truncation threshold T, a merging threshold U, and a maximum allowable number of GIW mixture components J_{max}

```
set $l = 0$ and $\boldsymbol{I} = \{i = 1, \dots, J_{k|k} | w_{k|k}^{(i)} > T\}$
repeat
    $l = l + 1$
    $j = \arg\max_{\mathrm{i}\in\mathbf{I}} w_{k|k}^{(i)}$
    $\boldsymbol{L} = \left\{i \in \boldsymbol{I} \middle| \left(m_{k|k}^{(i)} - m_{k|k}^{(j)}\right)^T \left(\hat{P}_{k|k}^{(j)}$[23]$\right)^{-1} \left(m_{k|k}^{(i)} - m_{k|k}^{(j)}\right) \leq U\right\}$
    $\tilde{w}_{k|k}^{(l)} = \sum_{i\in\boldsymbol{L}} w_{k|k}^{(i)}$
    $\tilde{\boldsymbol{m}}_{k|k}^{(l)} = \frac{1}{\tilde{w}_{k|k}^{(l)}} \sum_{i\in\boldsymbol{L}} w_{k|k}^{(i)} \boldsymbol{m}_{k|k}^{(i)}$
    $\tilde{P}_{k|k}^{(l)} = \frac{1}{\tilde{w}_{k|k}^{(l)}} \sum_{i\in\boldsymbol{L}} w_{k|k}^{(i)} P_{k|k}^{(i)}$ (without spread of means.)
    $\tilde{v}_{k|k}^{(l)} = \frac{1}{\tilde{w}_{k|k}^{(l)}} \sum_{i\in\boldsymbol{L}} w_{k|k}^{(i)} v_{k|k}^{(i)}$
    $\tilde{V}_{k|k}^{(l)} = \frac{1}{\tilde{w}_{k|k}^{(l)}} \sum_{i\in\boldsymbol{L}} w_{k|k}^{(i)} V_{k|k}^{(i)}$
    $\boldsymbol{I} = \boldsymbol{I} \backslash \boldsymbol{L}$
until $\boldsymbol{I} = \emptyset$
if $l > J_{max}$ then replace $\left\{\tilde{w}_{k|k}^{(i)}, \tilde{\boldsymbol{m}}_{k|k}^{(i)}, \tilde{P}_{k|k}^{(i)}, \tilde{v}_{k|k}^{(i)}, \tilde{V}_{k|k}^{(i)}\right\}_{i=1}^{l}$ by those of the $J_{max}$ GIW mixture components with largest weights
```

Output pruned and merged GIW mixture components $\left\{\tilde{w}_{k|k}^{(i)}, \tilde{\boldsymbol{m}}_{k|k}^{(i)}, \tilde{P}_{k|k}^{(i)}, \tilde{v}_{k|k}^{(i)}, \tilde{V}_{k|k}^{(i)}\right\}_{i=1}^{l}$ at time step k

[23] Compute $\hat{P}_{k|k}^{(j)}$ using (5.15).

A.5. MC simulation results of the parallel track scenario with 0.25m

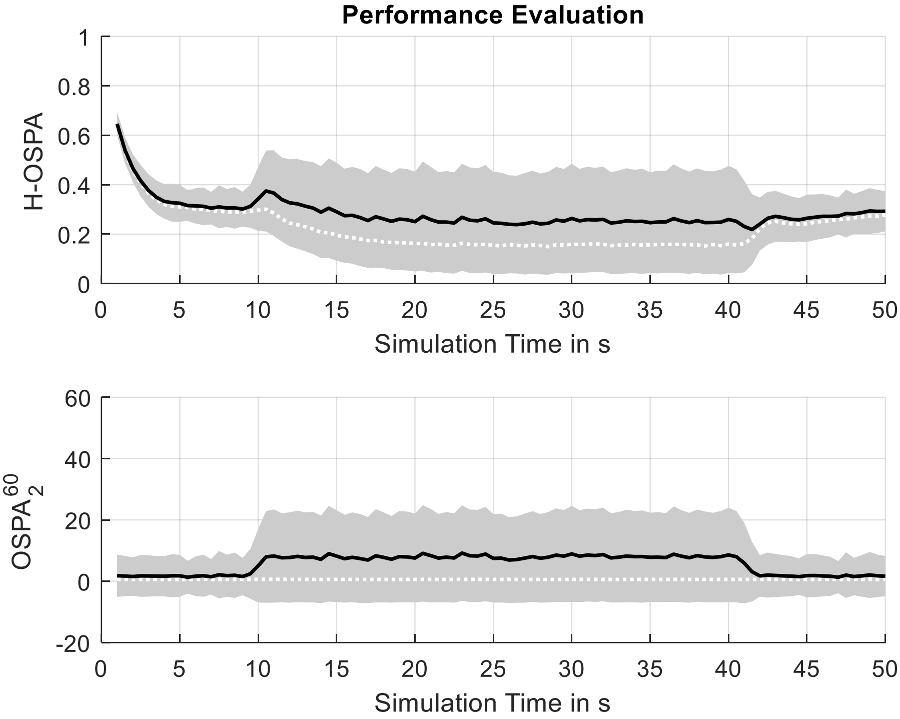

Figure 36: OSPA of DP at 0.25m distance between objects with $\Sigma = 0.5m \cdot s^{-1}$. The black line is the mean value with the grey area showing the +/- standard deviation range. The white dotted line gives the median value.

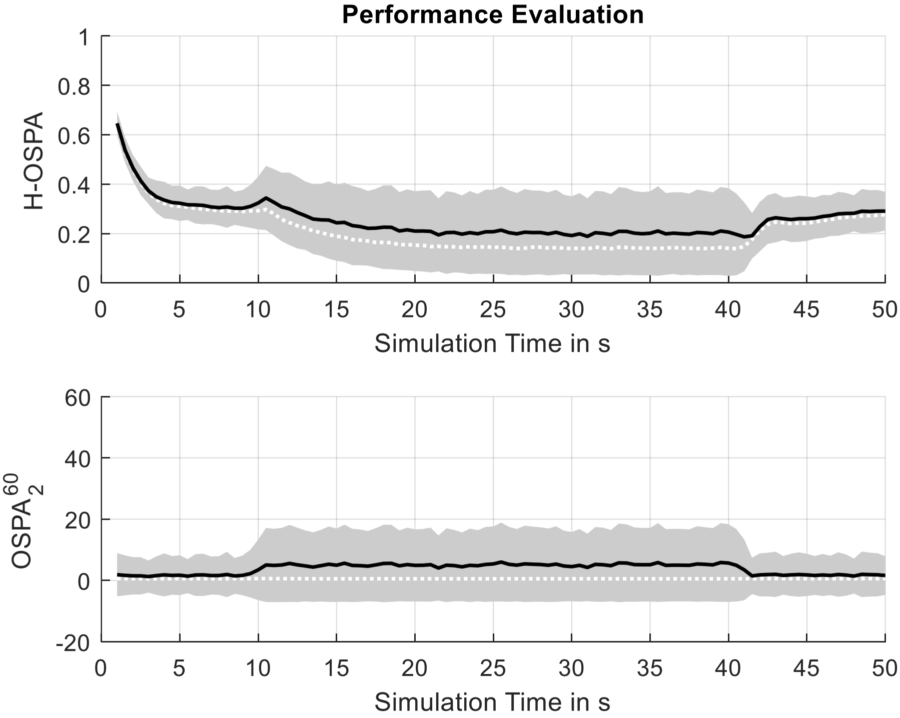

Figure 37: OSPA of DP with SP at 0.25m distance between objects with $\Sigma = 0.5m \cdot s^{-1}$. The black line is the mean value with the grey area showing the +/- standard deviation range. The white dotted line gives the median value.

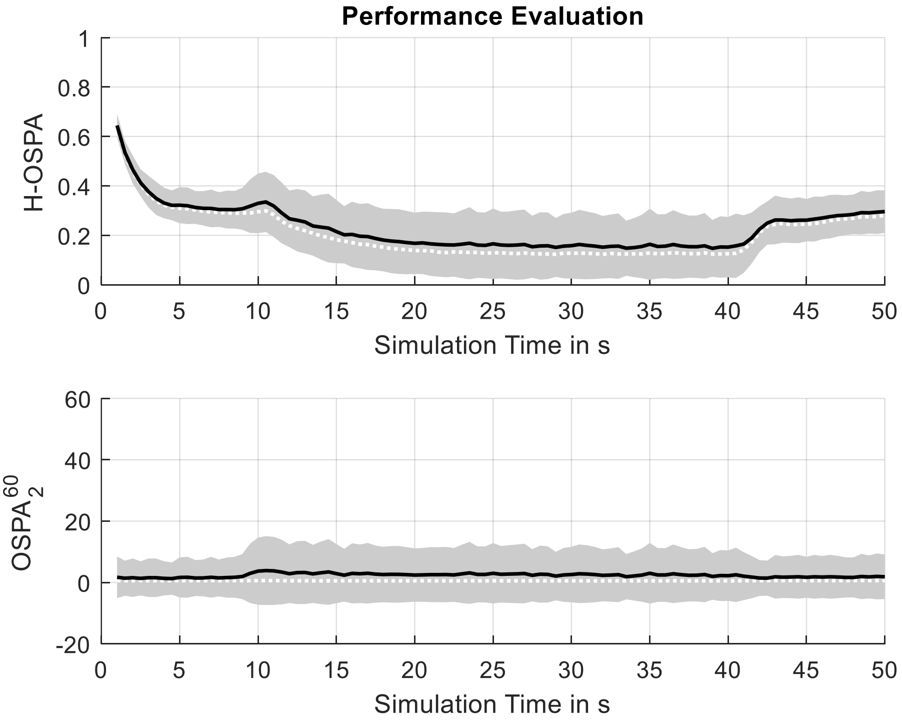

Figure 38: OSPA of StP at 0.25m distance between objects with $\Sigma = 0.5m \cdot s^{-1}$. The black line is the mean value with the grey area showing the +/- standard deviation range. The white dotted line gives the median value.

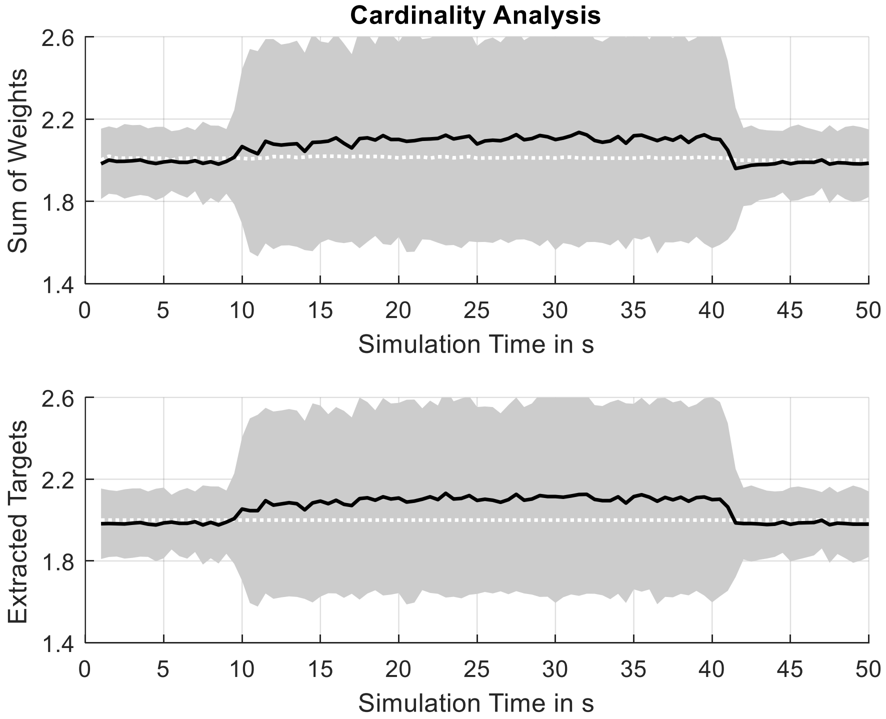

Figure 39: Cardinality evaluation of DP at 0.25m distance between objects with $\Sigma = 0.5m \cdot s^{-1}$. The black line is the mean value with the grey area showing the +/- standard deviation range. The white dotted line gives the median value.

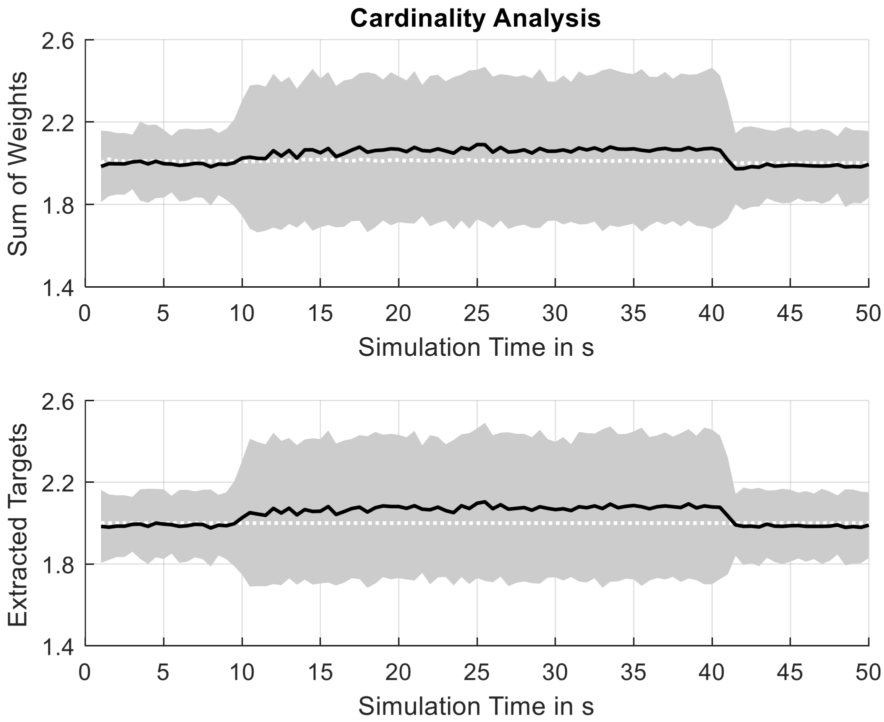

Figure 40: Cardinality evaluation of DP with SP at 0.25m distance between objects with $\Sigma = 0.5m \cdot s^{-1}$*. The black line is the mean value with the grey area showing the +/- standard deviation range. The white dotted line gives the median value.*

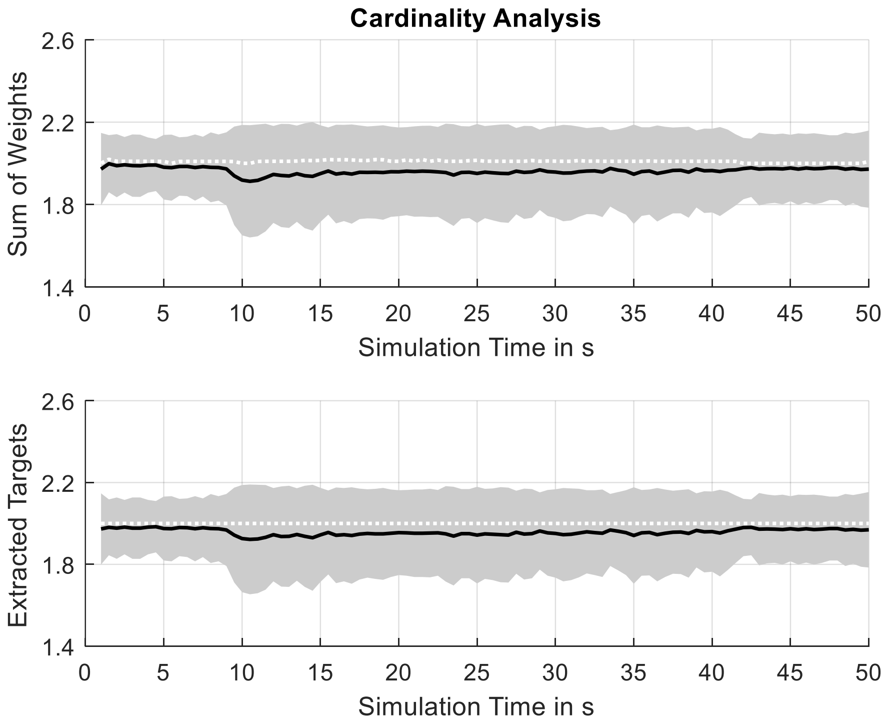

Figure 41: Cardinality evaluation of StP at 0.25m distance between objects with $\Sigma = 0.5m \cdot s^{-1}$*. The black line is the mean value with the grey area showing the +/- standard deviation range. The white dotted line gives the median value.*

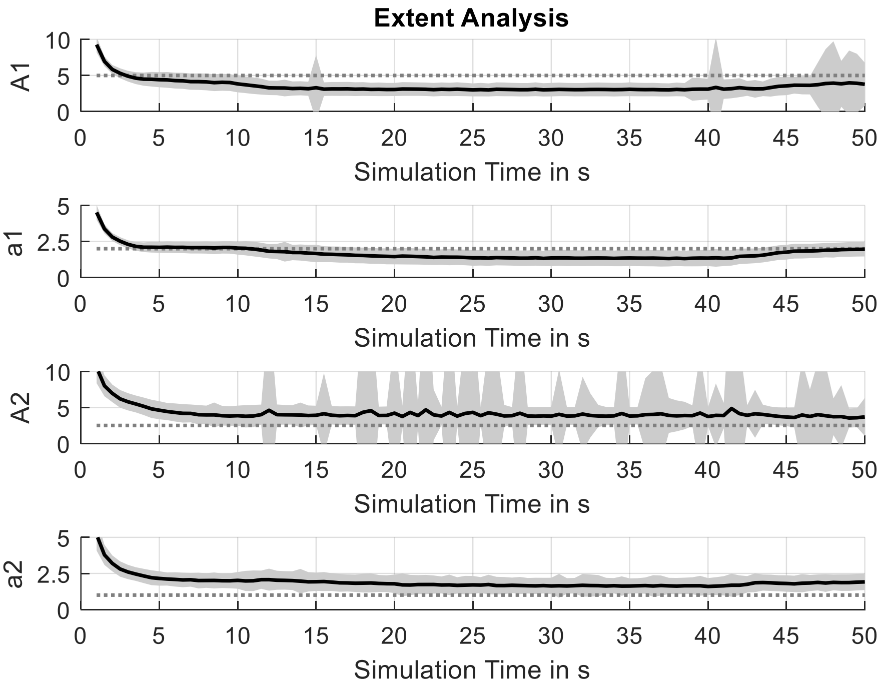

Figure 42: Half-axes estimation of DP at 0.25m distance between objects with $\Sigma = 0.5m \cdot s^{-1}$. The black line is the mean value with the grey area showing the +/- standard deviation range. The grey dotted line represents the true length of the half-axis in m.

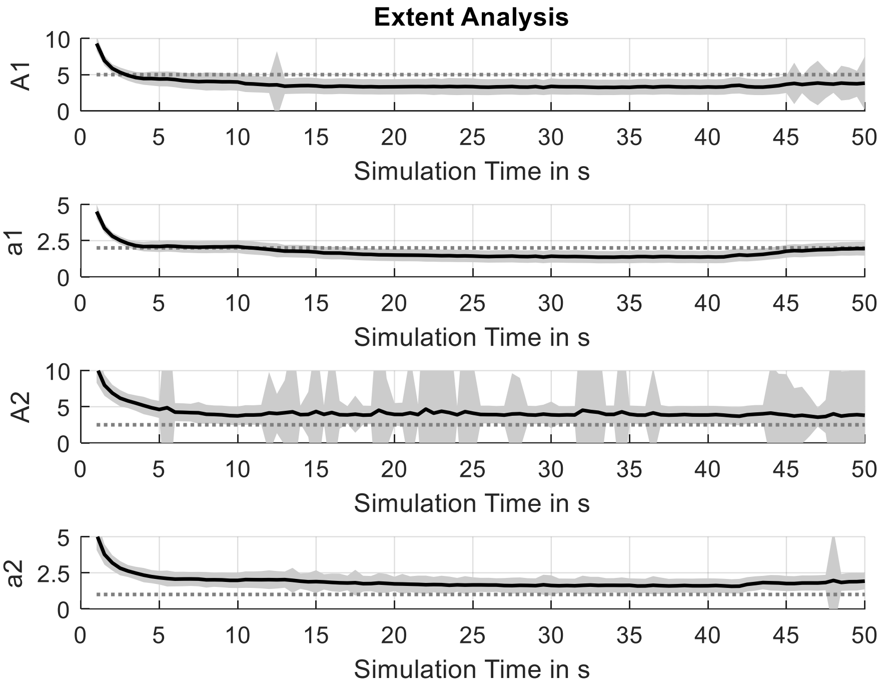

Figure 43: Half-axes estimation of DP with SP at 0.25m distance between objects with $\Sigma = 0.5m \cdot s^{-1}$. The black line is the mean value with the grey area showing the +/- standard deviation range. The grey dotted line represents the true length of the half-axis in m.

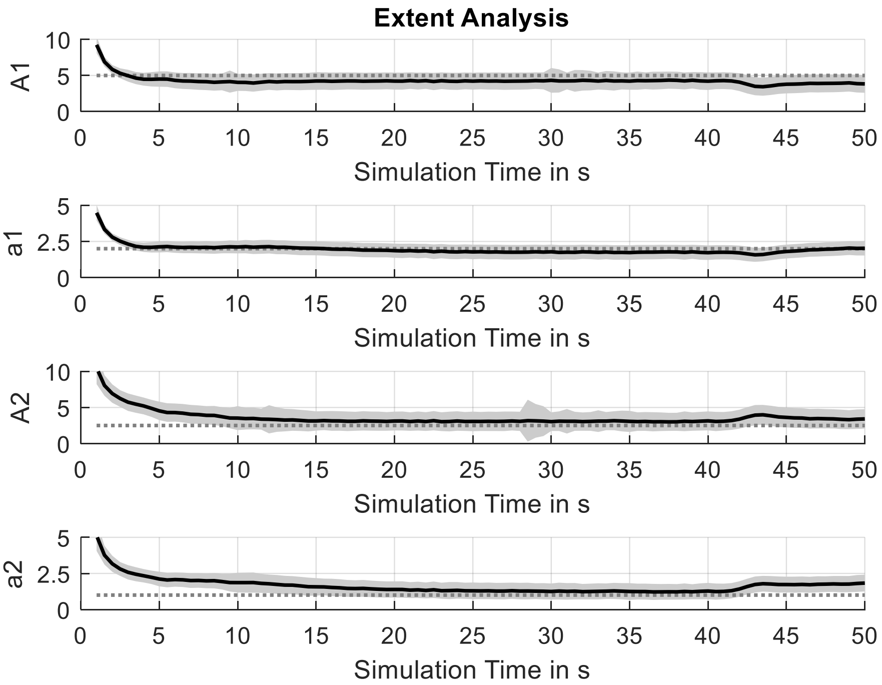

Figure 44: Half-axes estimation of StP at 0.25m distance between objects with $\Sigma = 0.5m \cdot s^{-1}$*. The black line is the mean value with the grey area showing the +/- standard deviation range. The grey dotted line represents the true length of the half-axis in m.*

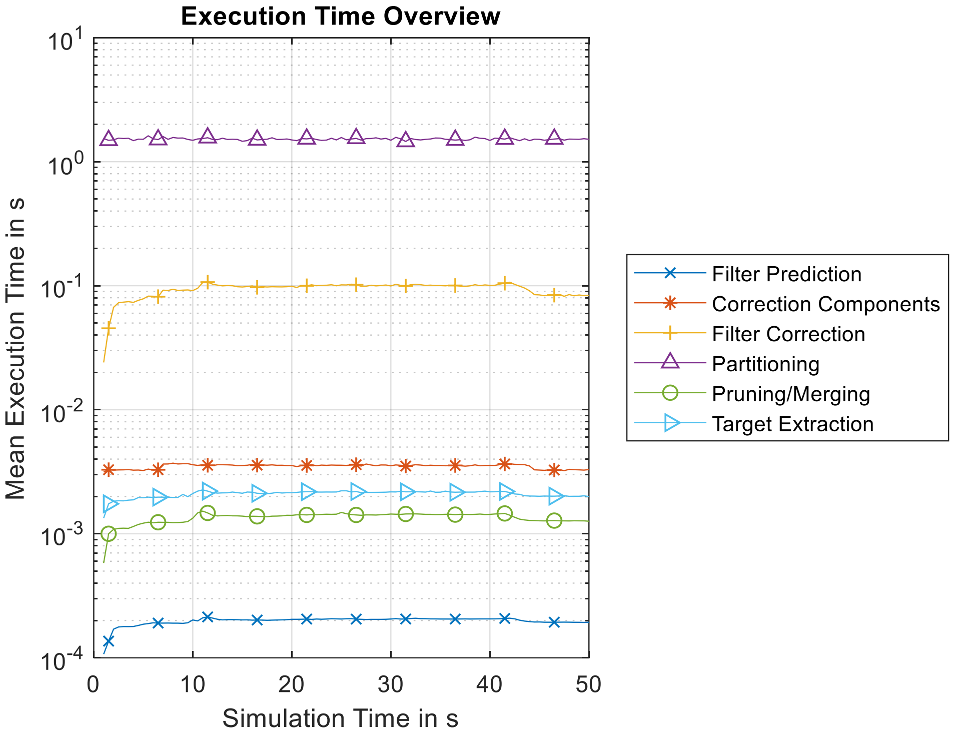

Figure 45: Averaged execution time of the single filter steps of DP at 0.25m distance between objects with $\Sigma = 0.5m \cdot s^{-1}$*.*

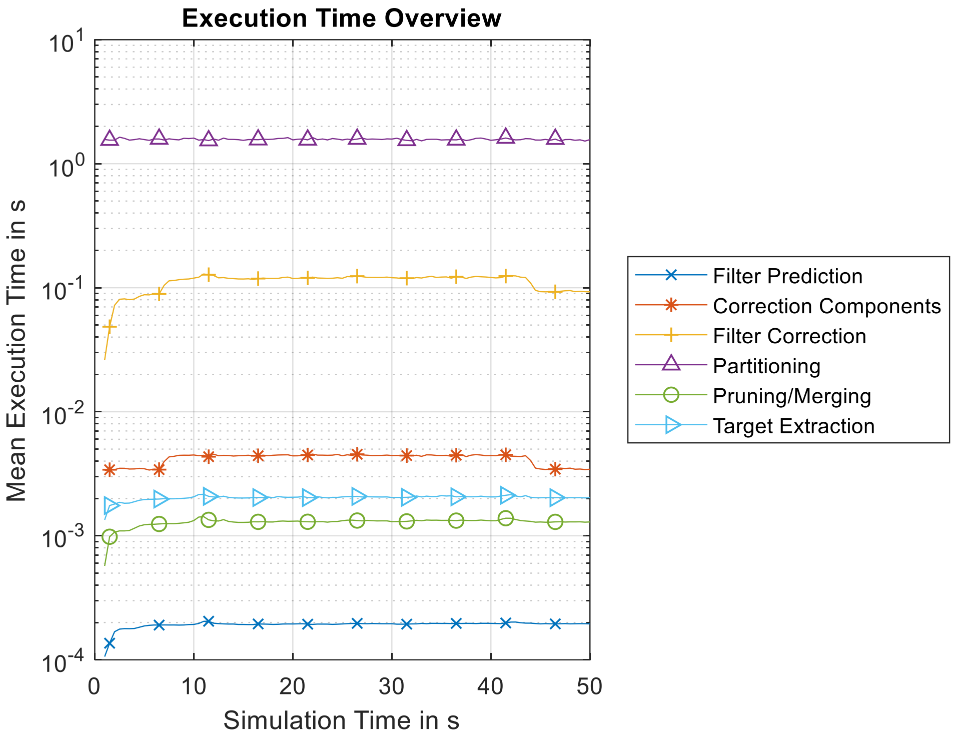

Figure 46: Averaged execution time of the single filter steps of DP with SP at 0.25m distance between objects with $\Sigma = 0.5m \cdot s^{-1}$.

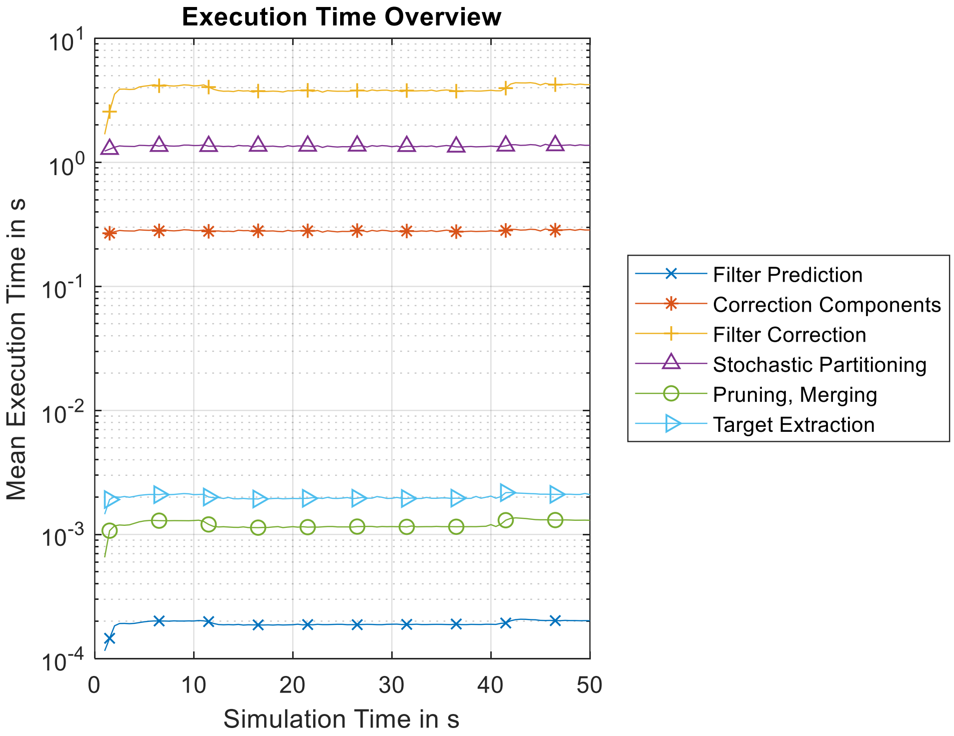

Figure 47: Averaged execution time of the single filter steps of StP at 0.25m distance between objects with $\Sigma = 0.5m \cdot s^{-1}$.

A.6. Execution time illustration separated into DP and SP

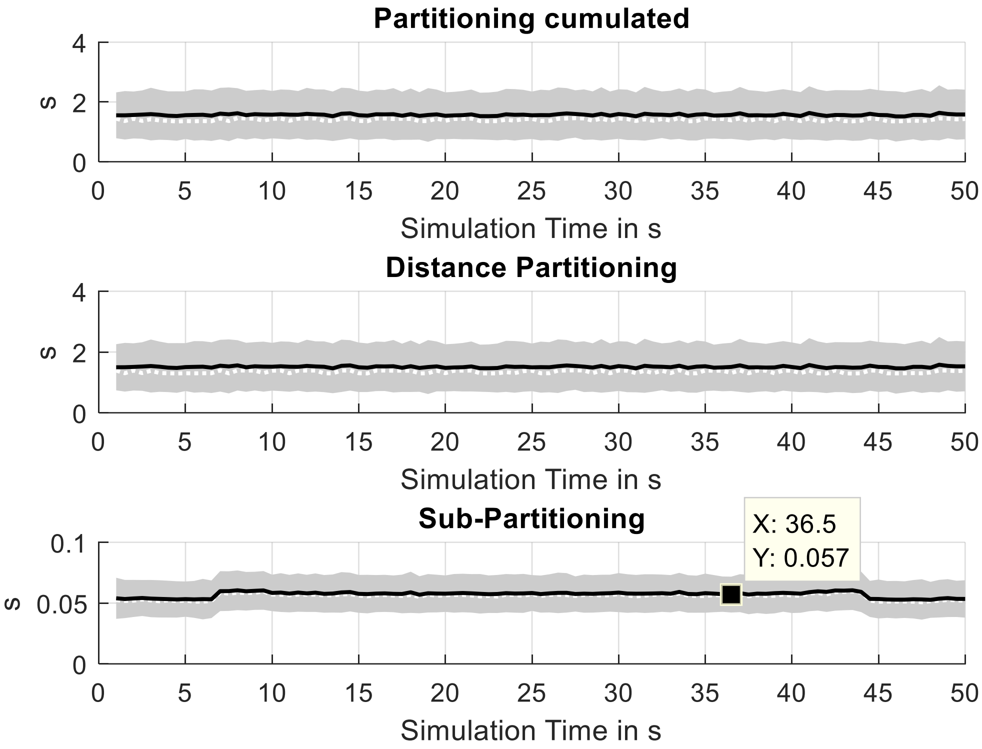

Figure 48: Execution time split to DP and SP. The upper sub-plot is the cumulated time of DP and SP. The influence of SP is not visible. In the middle sub-plot the time for DP is illustrated and in the lower sub-plot for SP. During the parallel phase of the objects the effort of SP increases.

VI. Bibliography

[1] Y. Bar-Shalom, Tracking and Data Association, San Diego, CA, USA: Academic Press Professional, Inc., 1987.

[2] M. Schuster, Multiple Object Tracking for Extended Targets using JIPDA filters, Aachen: Shaker Verlag, 2017, p. 224.

[3] R. P. S. Mahler, Statistical Multisource-Multitarget Information Fusion, Norwood, MA: Artech House, 2007.

[4] R. P. S. Mahler, ""Statistics 101" for Multisensor, Multitarget Data Fusion," *IEEE Aerospace and Electronic Systems Magazine,* vol. 19, pp. 53-64, 1 2004.

[5] T. Baur, "Tracking of Spline Modeled Extended Objects Using Random Finite Sets," *Master's Thesis. ISD, HTWG Konstanz,* pp. 1-79, 3rd April 2019.

[6] K. Granström and U. Orguner, "A PHD Filter for Tracking Multiple Extended Targets Using Random Matrices," *IEEE Transactions on Signal Processing,* vol. 60, pp. 5657-5671, 2012.

[7] J. W. Koch, "Bayesian Approach to Extended Object and Cluster Tracking using Random Matrices," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 44, pp. 1042-1059, 7 2008.

[8] R. P. S. Mahler, "PHD filters for nonstandard targets, I: Extended targets," in *2009 12th International Conference on Information Fusion*, 2009.

[9] K. Granström, C. Lundquist and O. Orguner, "Extended Target Tracking using a Gaussian-Mixture PHD Filter," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 48, pp. 3268-3286, 10 2012.

[10] K. Granström, L. Svensson, S. Reuter, Y. Xia and M. Fatemi, "Likelihood-Based Data Association for Extended Object Tracking Using Sampling Methods," *IEEE Transactions on Intelligent Vehicles,* vol. 3, pp. 30-45, 2018.

[11] R. P. S. Mahler, "PHD filters of higher order in target number," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 43, pp. 1523-1543, 10 2007.

[12] B.-N. Vo and W.-K. Ma, "The Gaussian Mixture Probability Hypothesis Density Filter," *IEEE Transactions on Signal Processing,* vol. 54, pp. 4091-4104, 11 2006.

[13] K. Gilholm and D. Salmond, "Spatial distribution model for tracking extended objects," *IEE Proceedings - Radar, Sonar and Navigation,* vol. 152, pp. 364-371, 10 2005.

[14] K. Granström, S. Renter, M. Fatemi and L. Svensson, "Pedestrian tracking using Velodyne data - Stochastic optimization for extended object tracking," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017.

[15] K. Granström, "Extended target tracking using PHD filters," *Linköping studies in science and technology. Dissertations. No. 1476,* pp. 1-405, 2012.

[16] X. Rong Li and V. P. Jilkov, "Survey of Maneuvering Target Tracking. Part I: Dynamic Models," *EEE Transactions on Aerospace and Electronic Systems,* vol. 39, pp. 1333-1364, 11 2003.

[17] X. Rong Li and V. P Jilkov, "A Survey of Maneuvering Target Tracking. Part III: Measurement Models," *Proceedings of SPIE Conference on Signal and Data Processingof Small Targets,* Vols. 4473-41, p. 24, 8 2001.

[18] R. P. S. Mahler, Advances in Statistical Multisource-Multitarget Information Fusion, Artech House, 2014.

[19] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of basic Engineering,* vol. 82, pp. 35-45, 1960.

[20] S. Challa, M. R. Morelande, D. Musicki and R. Evans, Fundamentals of object tracking, Cambridge: Cambridge University Press, 2011, p. 375.

[21] S. Thrun, W. Burgard and D. Fox, Probabilistic Robotics (Intelligent Robotics and Autonomous Agents), The MIT Press, 2005.

[22] S. J. Julier and J. K. Uhlmann, “A new extension of the Kalman filter to nonlinear systems,” in *Proc. of AeroSense: The 11th Int. Symp. on Aerospace/Defense Sensing, Simulations and Controls*, 1997.

[23] T. Lefebvre, H. Bruyninckx and J. De Schutter, “Kalman filters for nonlinear systems: A comparison of performance,” *International Journal of Control,* vol. 77, pp. 639-653, 2004.

[24] D. P. Bertsekas, “A distributed algorithm for the assignment problem,” *Laboratory for Information and Decision Sciences Working Paper at Massachusetts Institute of Technology, Cambridge,* pp. 1-19, 3 1979.

[25] H. W. Kuhn, “The Hungarian Method for the Assignment Problem,” 3 1955. [Online]. Available: https://doi.org/10.1002/nav.3800020109.

[26] Y. Bar-Shalom and E. Tse, “Tracking in a cluttered environment with probabilistic data association,” *Automatica,* vol. 11, pp. 451-460, 1975.

[27] B.-N. Vo, M. Mallick, Y. Bar-Shalom, S. Coraluppi, R. Osborne III, R. Mahler and B.-T. Vo, "Multitarget Tracking," in *Wiley Encyclopedia of Electrical and Electronics Engineering*, American Cancer Society, 2015, pp. 1-15.

[28] D. Musicki and R. Evans, “Joint Integrated Probabilistic Data Association: JIPDA,” *IEEE Transactions on Aerospace and Electronic Systems,* vol. 40, pp. 1093-1099, 8 2004.

[29] D. Musicki and B. La Scala, “Multi-Target Tracking in Clutter without Measurement Assignment,” *IEEE Transactions on Aerospace and Electronic Systems,* vol. 44, pp. 877-896, 8 2008.

[30] D. B. Reid, “An Algorithm for Tracking Multiple Targets,” *IEEE Transactions on Automatic Control,* vol. 24, pp. 843-854, 1 1980.

[31] R. P. S. Mahler, "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 39, pp. 1152-1178, 10 2003.

[32] R. P. S. Mahler, “"Statistics 102" for Multisource-Multitarget Detection and Tracking,” *IEEE Journal of selected Topics in Signal Processing,* vol. 7, 6 2013.

[33] B.-N. Vo, S. Singh and A. Doucet, “Sequential Monte Carlo Implementation of the PHD Filter for Multi-Target Tracking,” in *Proceedings of the 6th International Conference of Information Fusion*, 2003.

[34] K. Panta, D. E. Clark and B. Vo, "Data Association and Track Management for the Gaussian Mixture Probability Hypothesis Density Filter," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 45, pp. 1003-1016, 7 2009.

[35] K. Granström, M. Baum and S. Reuter, "Extended Object Tracking: Introduction, Overview, and Applications," *Journal of Advances in Information Fusion,* vol. 12, 12 2017.

[36] K. Gilholm, S. J. Godsill, S. Maskell and D. Salmond, "Poisson models for extended target and group tracking," *Proceedings of SPIE - The International Society for Optical Engineering,* vol. 5913, 8 2005.

[37] M. Baum, M. Feldmann, D. Fränken, U. Hanebeck and W. Koch, "Extended Object and Group Tracking: A Comparison of Random Matrices and Random Hypersurface Models.,"

in *Neue Perspektiven für die Informatik, Beiträge der 40. Jahrestagung der Gesellschaft für Informatik e.V. (GI)*, 2010.

[38] A. K. Gupta and D. K. Nagar, Matrix Variate Distributions, Vols. Monographs and Surveys in Pure and Applied Mathematics, Book 104, USA: Chapman and Hall/CRC, 1999.

[39] G. Keuk and S. S. Blackman, “On phased-array radar tracking and parameter control,” *IEEE Transactions on Aerospace and Electronic Systems,* vol. 29, pp. 186-194, 1 1993.

[40] M. Feldmann, D. Fränken and W. Koch, "Tracking of Extended Objects and Group Targets Using Random Matrices," *IEEE Transactions on Signal Processing,* vol. 59, pp. 1409-1420, 4 2011.

[41] M. Feldmann and D. Fränken, "Advances on Tracking of Extended Objects and Group Targets using Random Matrices," in *12th International Conference on Information Fusion*, 2009.

[42] K. Granström, C. Lundquist and U. Orguner, "A Gaussian mixture PHD filter for extended target tracking," in *2010 13th International Conference on Information Fusion*, 2010.

[43] K. Granström and U. Orguner, "Implementation of the GIW-PHD filter," 2012.

[44] D. P. Bertsekas, “The auction algorithm: A distributed relaxation method for the assignment problem,” *Annals of Operations Research,* vol. 14, pp. 105-123, 01 12 1988.

[45] K. G. Murty, “An Algorithm for Ranking all the Assignments in Order of Increasing Cost,” *Operations Research,* vol. 16, pp. 682-687, 1968.

[46] G.-C. Rota, “The Number of Partitions of a Set,” *The American Mathematical Monthly,* vol. 71, pp. 498-504, 1964.

[47] K. Granström, U. Orguner, R. Mahler and C. Lundquist, “Corrections on: Extended Target Tracking Using a Gaussian-Mixture PHD Filter,” *IEEE Transactions on Aerospace and Electronic Systems,* vol. 53, pp. 1055-1058, 4 2017.

[48] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Berlin, Heidelberg: Springer-Verlag, 2006.

[49] M. Ester, H.-P. Kriegel, J. Sander and X. Xu, “A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Portland, 1996.

[50] M. Ankerst, M. M. Breunig, H.-P. Kriegel and J. Sander, “OPTICS: Ordering Points to Identify the Clustering Structure,” in *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, New York, NY, USA, 1999.

[51] Y. Zhang and H. Ji, "Robust bayesian partition for extended target gaussian inverse wishart PHD filter," *IET Signal Processing,* vol. 8, pp. 330-338, 6 2014.

[52] H. Zhu, P. Zhang and T. Ma, "Research on measurement set partitioning method for tracking multiple extended targets," in *27th Chinese Control and Decision Conference (CCDC)*, 2015.

[53] P. Li, H. Ge, J. Yang and H. Zhang, "Shape selection partitioning algorithm for Gaussian inverse Wishart probability hypothesis density filter for extended target tracking," *IET Signal Processing,* vol. 10, pp. 1041-1051, 2016.

[54] D. Ravenzwaaij, P. Cassey and S. D. Brown, “A simple introduction to Markov Chain Monte Carlo sampling,” *Psychonomic Bulletin & Review,* vol. 25, pp. 143-154, 01 2 2018.

[55] S. Jain and R. M. Neal, “A Split-Merge Markov Chain Monte Carlo Procedure for the Dirichlet Process Mixture Model,” *Journal of Computational and Graphical Statistics,* vol. 13, 8 2000.

[56] R. M. Neal, “Markov Chain Sampling Methods for Dirichlet Process Mixture Models,” *Journal of Computational and Graphical Statistics,* vol. 9, pp. 249-265, 2000.

[57] K. Granström, M. Fatemi and L. Svensson, “Gamma Gaussian inverse-Wishart Poisson multi-Bernoulli filter for extended target tracking,” in *19th International Conference on Information Fusion (FUSION)*, 2016.

[58] K. Granström, M. Fatemi and L. Svensson, “Poisson multi-Bernoulli conjugate prior for multiple extended object estimation,” [Online]. Available: https://arxiv.org/abs/1605.06311. [Accessed 13 04 2019].

[59] D. Schuhmacher, B. Vo and B. Vo, "A Consistent Metric for Performance Evaluation of Multi-Object Filters," *IEEE Transactions on Signal Processing,* vol. 56, pp. 3447-3457, 8 2008.

[60] S. Nagappa, D. E. Clark and R. Mahler, "Incorporating track uncertainty into the OSPA metric," in *14th International Conference on Information Fusion*, 2011.

[61] K. Granström and U. Orguner, "On Spawning and Combination of Extended/Group Targets Modeled With Random Matrices," *IEEE Transactions on Signal Processing,* vol. 61, pp. 678-692, 2 2013.

[62] K. Granström and U. Orguner, “Estimation and maintenance of measurement rates for multiple extended target tracking,” in *2012 15th International Conference on Information Fusion*, 2012.