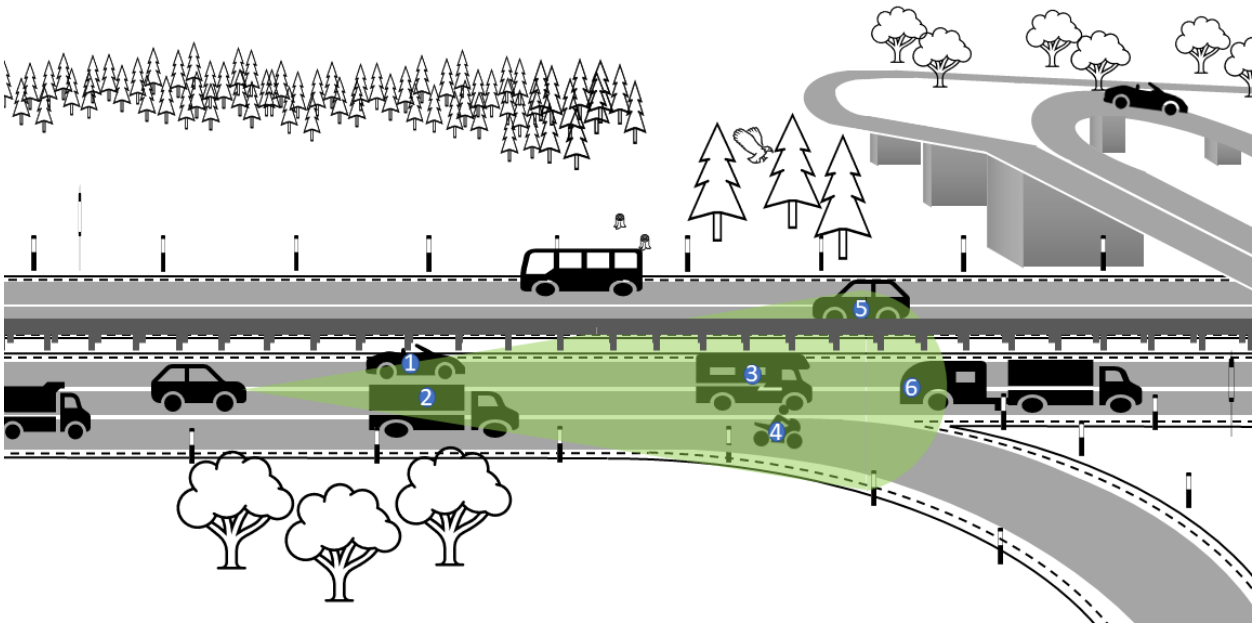
CHALMERS
UNIVERSITY OF TECHNOLOGY



Extended Multiple Target Tracking for Automotive Applications

Master's thesis in Systems, Control and Mechatronics

Evelina Dahlsten & Mahandokht Rafidashti

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021

Master's thesis 2021

Extended Multiple Target Tracking for Automotive Applications

Evelina Dahlsten & Mahandokht Rafidashti

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2021

Extended Multiple Target Tracking
for Automotive Applications
Evelina Dahlsten & Mahandokht Rafidashti

Academic supervisor & examiner: Lennart Svensson, Department of Electrical Engineering
Advisor: Maryam Fatemi, Zenseact

Master's Thesis 2021
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: An illustration of a highway scenario where an ego vehicle with a single front radar tracks the vehicles in the field of view (2D green area) of the front radar.

Typeset in LaTeX
Gothenburg, Sweden 2021

Extended Multiple Target Tracking for Automotive Applications
Evelina Dahlsten & Mahandokht Rafidashti
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Target tracking is a collection of methods that, in automotive applications, includes the tracking of vehicles, pedestrians, and other road users. Commonly, the targets' positions are estimated, but the tracker can also include information such as the velocity and acceleration of the tracked targets. Target tracking is a necessity for autonomous vehicles, where the idea is to use the tracking information to make decisions. The field is constantly developing and new methods are regularly being tried and evaluated. The reliability of target tracking algorithms is critical since the vehicle relies on information gained from them to make decisions; in critical traffic scenarios, it can be a matter of life and death for the passengers.

This thesis focuses on Extended Multiple Target Tracking (EMTT), where the number of targets is unknown and each target can generate several measurements. Poisson Multi-Bernoulli Mixture (PMBM) based filters have been proven to yield state-of-the-art results. These filters are thus studied and evaluated in this thesis based on their suitability for tracking vehicles on the road using radar data. The thesis is only considering highway scenarios, therefore slow-moving traffic or stop-and-go scenarios are omitted. Three variations of the Poisson Multi-Bernoulli (PMB) filter were implemented, the Track Oriented PMB (TO-PMB), the Most Likely Assignment PMB (MLA-PMB), and the Efficient Approximation of Feasible Set PMB (EAFS-PMB). Two different birth models were also implemented for the three filters, a uniform birth model and a measurement-driven birth model. The results for the three different filters were rather similar, where the main difference was observed when changing the birth model. The uniform birth model tends to be more accurate over time, while the measurement-driven birth model limits the computational complexity and performs tracking approximately 3 to 8 times faster than the uniform birth model, depending on the complexity of the scenario.

Keywords: Extended Multi-Object Tracking, EMOT, Extended Multi-Target Tracking, EMTT, Poisson Multi-Bernoulli Mixture, PMBM, Poisson Multi-Bernoulli, PMB, Radar, Autonomous Vehicles

Acknowledgements

Throughout writing this master's thesis, we have received a great deal of support and assistance. First of all, we would like to thank our supervisor Maryam Fatemi at Zenseact for always taking time for discussions to give us perspective on the implementation and results, and giving us insightful feedback throughout the writing process of this thesis.

We would also like to thank our supervisor and examiner at Chalmers University of Technology, Lennart Svensson, for encouraging us to maximize the outcome of this thesis and to his Ph.D. student Yuxuan Xia for giving us valuable insight into the algorithms and specifics in theory and implementation.

Furthermore, we would like to thank our peers for insightful feedback regarding both the structure and content of this work, you know who you are. We would also like to thank everyone who's helped us along the way to make this journey possible.

Evelina and Mahandokht, June 2021

Contents

List of Figures

List of Tables

List of Acronyms

ADAS	Advanced Driver Assistance Systems
CA	Constant Acceleration
CPHD	Cardinalized Probability Hypothesis density
CT	Coordinated Turn
CV	Constant Velocity
DBSCAN	Density-Based Spatianl Clustering of Applications with Noise
GGIW	Gamma Gaussian Inverse Wishart
GH	Global Hypothesis
GHLT	Global Hypothesis Lookup Table
GIW	Gaussian Inverse Wishart
GLMB	Generalized Label Multi Bernoulli
IMU	Inertial Measurement Unit
KLD	Kullback Leibler Divergence
LH	Local Hypothesis
LMB	Labelled Multi Bernoulli
MB	Multi Bernoulli
MBM	Multi Bernoulli Mixture
MHT	Multi Hypothesis Tracking
MOT	Multiple Object Tracking
MTT	Multiple Target Tracking
PDF	Probability Density Function
PHD	Probability Hypothesis density
PMB	Poisson Multi Bernoulli
PMBM	Poisson Multi Bernoulli Mixture
PPP	Poisson Point Process
RFS	Random Finite Set

1
Introduction

The interest in autonomous vehicles has grown strong during the past years, and the reasons for this are many, including safety, comfort, and driver convenience. In 2015, the U.S. Department of Transportation stated that 94 % of all lightweight vehicle crashes have the driver as the immediate reason for the critical pre-crash event [2]. The driver-related critical reasons could further be assigned to mostly recognition and decision errors, but also performance errors and non-performance errors, including sleepy drivers. Developing autonomous vehicles that keep track of their surroundings, perform object tracking, and make calculation-based decisions, can be quite beneficial for traffic safety. The concept of autonomous vehicles is under development and so far, most advances are related to automated functions and the vehicle's advanced driver-assistance system (ADAS). The equipment used for these functions is therefore varying. Examples of measuring equipment are radars, LIDARs, high-quality cameras, GPS, and IMUs, where this thesis will make use of radar detections.

To make a vehicle autonomous, many functionalities need to be implemented, including algorithms to track other vehicles and pedestrians on the road. This study is done in collaboration with Zenseact and focuses on Multiple Target Tracking (MTT), also called Multiple Object Tracking (MOT), which is a technique used to track the movement and dynamic properties of multiple objects. Objects and targets are the same thing and will be used interchangeably in this thesis. MTT is a subject that has significantly developed during the past years. In the context of autonomous driving, MTT is a collection of methods that includes tracking vehicles, pedestrians, bicyclists, and other road users. To increase safety, the idea is to first do tracking and then use methods for predicting the movement of other objects and use that information to slow down or emergency brake the vehicle if needed.

In MTT, there are two different ways to model objects. One way is to assume point objects, where, if considering a two-dimensional plane, a point object is an object that has no length or width, and can be described only by its kinematics. This is used in many applications of multi-target tracking, such as in radar-based air surveillance where the targets are far away. In these cases, the targets are assumed to be points in the field of view. This assumption is not viable when tracking vehicles on the road with vehicle-mounted radars, since the targets in this application are rather close to each other and each target can generate several measurement points at each time step. Here, the objects are therefore handled as extended objects, which means that several measurement points can be assigned to one object, and

the shape of the target, called the extent of the target, is unknown. The number of detections from one object depends on the distance to the object, the extent of the object, and the sensor resolution. In this case, the estimation of the target extent should be included when solving the tracking problem.

Within the field of MTT, there are several different filtering methods used, including Multi Hypothesis Tracking (MHT) [3], Probability Hypothesis Density (PHD) filter [4], Multi-Bernoulli Mixture (MBM) filter [5], and Poisson Multi-Bernoulli Mixture (PMBM) filter [6], where the PMBM filter has been shown to have state-of-the-art performance. However, the PMBM filters are computationally complex filters. To reduce the complexity, Poisson Multi bernoulli (PMB) filters were developed in [7, 1] as a variation of the PMBM filter, where the complexity is reduced by reduction methods and approximations. In the literature, the PMB filters have been evaluated for simulated data, and there is thus a gap in the literature for how it performs using real data. This study focuses on filtering methods based on the PMBM filter, where the PMB filter is implemented using measurements collected with radars mounted on a moving vehicle to track other vehicles on the road. Furthermore, when using real data, the number of measurements at each time step will be significantly high which creates a need for good reduction methods that retain as much info as possible.

1.1 Research Questions

The main objective of this thesis is to choose three PMBM-based filters by doing a literature study. The filters are thereafter implemented to solve the multiple extended target tracking problem for automotive applications, i.e., to track the movement of vehicles on the road.

The general research questions for this work are:

- Which PMBM-based filters are more appropriate for solving the MTT problem in automotive applications (e.g., to track vehicles on the road)?
- What are the advantages and disadvantages of each filter?
- How does the target birth model affect the filter performance?

1.2 Scope

In this work, the filtering problem is solved by performing prediction and update using data up to and including the current time step. Smoothing or estimation of trajectories is not included. Further, no change to the environment is made, and the algorithm will not be deployed in any vehicle. The run times, however, need to be manageable in terms of computational complexity. Therefore, appropriate reduction methods are being used.

In automotive applications, there are a lot of different traffic scenarios that can occur. To make this thesis feasible, we are focusing on highway scenarios. We are

thus not considering slow-moving traffic or stop-and-go scenarios. Several highway sequences are used for tuning the filter, and three sequences with different scenarios are further investigated and evaluated. In this work, all the objects are assumed to move in a two-dimensional plane.

1.3 Related Work

According to [8], some of the more advanced filtering methods used to deal with multiple extended target tracking are Generalized Labelled Multi-Bernoulli (GLMB) [9], Labelled Multi-Bernoulli Filter (LMB) [10], Multi-Bernoulli Mixture (MBM) [5] and Poisson Multi-Bernoulli Mixture (PMBM) based methods [6]. These methods are briefly described in the following paragraphs.

The Generalized Labelled Multi-Bernoulli (GLMB) filter [9] solves the MTT problem using the standard point-detection likelihood model based on a type of labeled Random Finite Set (RFS). In labeled random finite sets, a specific label is assigned to each target so that its trajectory can be easily identified without the need for post-processing [9]. In [9], the GLMB filter was compared to an extended target Probability Hypothesis Density (PHD) filter [4] and an extended target Cardinalized Probability Hypothesis Density (CPHD) filter [11], both of which had a worse performance than the GLMB. The article found that the number of posterior components grows exponentially, so a better alternative would be the Labelled Multi-Bernoulli (LMB) filter, which is an approximation of the GLMB filter. According to [12], LMB can estimate tracks, is unbiased in posterior cardinality, and has a lower computational complexity without losing the desirable qualities of GLMB.

The PMBM conjugate prior is a linear combination of independent Poisson Point Process (PPP) and Multi-Bernoulli Mixture components which was developed for point target MTT in [13]. The MBM and PMBM filters are very similar, where the MBM is a special case of the PMBM where the PPP is neglected as a birth model by setting its intensity to 0 and object birth is modelled as a Multi-Bernoulli mixture density [5]. The authors of [5] found that the MBM filter fails to estimate a state for each target if the number of newborn targets that are detected is higher than the number of birth components. The article also states that if the birth location is vague, the PMBM filter will yield better performance than the MBM filter.

The PMBM filter has a more efficient filter structure and is less computationally complex compared to the GLMB filter since it models the birth as a Poisson Point Process. This enables quick detection of targets born close to each other as well as the use of recycling methods, which are used to improve performance. The PMB filter, which is an approximation of the PMBM filter, has the best overall performance considering the accuracy of the estimations and the computational complexity [13].

PMBM-based filters have generally been proven to yield great results [1, 14], and the filtering method is described in detail in Chapter 2. Previous comparable studies have used simulated data. During this project, a new study using three-dimensional

lidar data with a PMBM filter was published [15]. This study explores the application of PMB filters on two-dimensional radar data, which is why these studies are not comparable.

1.4 Main Contributions

In this thesis, the main contributions can be summarized as:

- We select and implement a set of different PMB filters that have the potential to yield good performance. These filters are described regarding how they can be used in Extended Multi Object Tracking (EMOT).

- We analyze the performance of the PMB filters on real data. In particular, we compare the results for three PMB filters, namely the Track Oriented (TO) PMB, the Most Likely Assignment (MLA) PMB and the Efficient Approximation of Feasible Set (EAFS) PMB. The implementation is done for highway scenarios with real two-dimensional radar data, from a radar mounted on a vehicle.

- We implement two different birth models and compare the results for all three filters using these models.

2
Theory

This section presents the literature needed to understand the steps in this project. We first describe filtering in general and the Kalman filter for point targets, where one target is tracked at a time. This is used in the prediction and update step for every object and filter recursion, together with a motion and measurement model, respectively. We then explain Extended Multiple Target Tracking (EMTT) and the added complexity it entails, where the data association problem is described. Further, the Poisson Multi-Bernoulli Mixture filter is described. The filters which are implemented in this thesis, meaning the different variations of the Poisson Multi-Bernoulli filter, are also presented in more detail. Finally, the used evaluation metric, the Generalized Optimal Sub-Pattern Assignment (GOSPA) metric, is explained for extended objects.

In this work, scalar variables are represented by lowercase letters such as x and vectors are denoted by bold lowercase letters such as $\mathbf{x}$. Uppercase letters such as X and bold uppercase letters such as $\mathbf{X}$ denote matrices and sets of vectors, respectively.

2.1 Bayesian Filtering

Target tracking algorithms are based on solving the Bayesian filtering problem. The Bayesian filtering problem consists of recursively estimating the probability density function of the state of an object (the posterior) given the previous state of the object (the prior) and measurements at each time step. The filtering problem can be solved by using Bayesian statistics.

2.1.1 Bayesian Statistics

From a philosophical point of view, there are two main approaches to statistics: the frequentist or traditional approach and the Bayesian approach. Traditional statistics introduces probabilities for random events which result from random experiments, so probability is defined as the relative frequency with which an event occurs given many experiments [16]. In this work the Bayesian approach will be used, which considers the variables to be random. One can then directly use the rules of probability, which follow logical and consistent reasoning, to predict parameters and revise our predictions given new information [16, 17].

In Bayesian statistics, there is an initial probability statement called the prior probability which can be combined with observed data to obtain the updated or posterior probability. The distribution of a continuous random variable is represented by a probability density function (PDF) and it is denoted by $p(\theta)$ where θ is the random variable. One can also show the distribution of a random variable as $\theta \sim p(\theta)$. The PDF gives us the probability of a random variable falling within a certain range of values. Consider the variable θ which is a random variable with known prior distribution $p(\theta)$ and the measurements $y_{1:T}$ which are also random variables. Here, $y_{1:T}$ denotes a sequence of measurements from time step 1 to time step T. Assuming that the conditional distribution $p(y_{1:T}|\theta)$ is known, one can find the posterior distribution of θ denoted by $p(\theta|y_{1:T})$ using Bayes' rule:

$$p(\theta|y_{1:T}) = \frac{p(y_{1:T}|\theta)p(\theta)}{p(y_{1:T})} \propto p(y_{1:T}|\theta)p(\theta), \tag{2.1}$$

where $p(y_{1:T})$ can be considered a normalization term which is independent of θ [18].

Random variables can be represented by several different probability distributions. One of the most commonly used distributions is the univariate Gaussian or normal distribution, the pdf of which can be written as:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\Big(-\frac{1}{2}\big(\frac{x-\mu}{\sigma}\big)^2\Big), \tag{2.2}$$

where, μ is the mean and σ is the standard deviation. These are the main parameters of a normal distribution and a Gaussian-distributed random variable can be defined entirely by them. One can use the notation $x \sim \mathcal{N}(x; \mu, \sigma^2)$ to show that a variable is Gaussian with parameters μ and σ [19]. For a vector $\mathbf{x} = \{x_1, x_2, \ldots x_k\}$ such that $\mathbf{x} \sim \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ where $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{\Sigma}$ is the covariance matrix, the multivariate normal distribution $p(\mathbf{x})$ can be written as [20]:

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^k|\boldsymbol{\Sigma}|}} \exp\Big(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\Big), \tag{2.3}$$

where $|\boldsymbol{\Sigma}|$ denotes the determinant of the matrix $\boldsymbol{\Sigma}$. Other common distributions such as the uniform distribution, the Gamma distribution, and the inverse Wishart distribution will be discussed in the following sections.

As mentioned previously, filtering consists of recursively estimating the state of an object. These recursions are performed in two steps: the prediction step and the update step. The target state vector at time step k is denoted by $\mathbf{x}_k$ and the measurement vectors from the first time step to time step k is denoted by $\mathbf{z}_{1:k}$. The prediction is done using the Chapman-Kolmogorov equation:

$$p(\mathbf{x}_{k+1}|\mathbf{z}_{1:k}) = \int p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k})d\mathbf{x}_k\,, \tag{2.4}$$

to find the predicted density $p(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})$ and the update is performed using Bayes' update:

$$p(\mathbf{x}_{k+1}|\mathbf{z}_{1:k+1}) \propto p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})p(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})\,, \tag{2.5}$$

to find the posterior density $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. In the above equations, $p(\mathbf{x}_k|\mathbf{z}_{1:k})$, $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ and $p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})$ are the prior density, the transition density and the measurement likelihood, respectively. The transition density describes the relation between the current object state with the previous state, and the measurement likelihood describes the relation between the measurement at time step k and the target state at that time.

2.1.2 Kalman Filter

The filtering equations (2.4) and (2.5) apply to all filtering problems, but there is no general analytical expression for the posterior. There are few examples where the posterior has an analytical expression given certain models or assumptions. The Kalman filter computes the analytical solution to the filtering problem for linear and Gaussian models. The general form of the linear and Gaussian state-space model is shown below for state vector $\mathbf{x}_k \in \mathbb{R}^n$ and measurement $\mathbf{z}_k \in \mathbb{R}^m$:

$$\mathbf{x}_{k+1} = F_k\mathbf{x}_k + \mathbf{q}_k, \tag{2.6a}$$
$$\mathbf{z}_{k+1} = H_{k+1}\mathbf{x}_{k+1} + \mathbf{r}_{k+1}, \tag{2.6b}$$
$$\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_0, P_{0|0}), \tag{2.6c}$$
$$\mathbf{q}_k \sim \mathcal{N}(\hat{\mathbf{q}}_k, Q_k), \tag{2.6d}$$
$$\mathbf{r}_{k+1} \sim \mathcal{N}(\hat{\mathbf{r}}_{k+1}, R_{k+1}), \tag{2.6e}$$

where, (2.6a) is the motion model where $F_k \in \mathbb{R}^{n\times n}$ is the transformation matrix and $\mathbf{q}_k$ is the process noise. The measurement model is described in (2.6b), where $H_{k+1} \in \mathbb{R}^{m\times n}$ is the measurement model matrix and $\mathbf{r}_{k+1}$ is the measurement noise. Motion models, also called process models, describe the kinematics of the objects that are being tracked and how these objects are expected to move over time. The measurements generated from these objects are described together with their properties by the measurement model, which is dependent on the sensor used to collect them. In linear and Gaussian models the prior $\mathbf{x}_0$ is Gaussian with mean $\hat{\mathbf{x}}_0$ and covariance $P_{0|0}$, and so are the motion and measurement noise (2.6d), (2.6e), where Q_k and R_{k+1} are the process noise and measurement noise covariance matrices, respectively. The noise distributions are also independent of each other and in time. This means that the predicted density,

$$p(\mathbf{x}_{k+1}|\mathbf{z}_k) = \mathcal{N}(\mathbf{x}_{k+1}; \hat{\mathbf{x}}_{k+1|k}, \bar{P}_{k+1|k}), \tag{2.7}$$

and the posterior density,

$$p(\mathbf{x}_{k+1}|\mathbf{z}_{1:k+1}) = \mathcal{N}(\mathbf{x}_{k+1}; \hat{\mathbf{x}}_{k+1|k+1}, \bar{P}_{k+1|k+1}), \tag{2.8}$$

for these models will always be Gaussian, so it is enough to compute the moments $\hat{\mathbf{x}}_{k+1|k}$, $P_{k+1|k}$, $\hat{\mathbf{x}}_{k+1|k+1}$ and $P_{k+1|k+1}$ for the predicted and posterior densities at each time step to estimate the state of an object at that time. Here, the notation "$k+1|k$" means the value at time step $k+1$ given measurements from the first time step up

to and including time step k.

Assuming that the motion and measurement noises are zero mean, the mean and covariance of the predicted density are calculated as:

$$\begin{aligned} \hat{\mathbf{x}}_{k+1|k} &= F_k \hat{\mathbf{x}}_{k|k}, \\ P_{k+1|k} &= F_k P_{k|k} F_k^T + Q_k, \end{aligned} \tag{2.9}$$

and the update is calculated as:

$$\begin{aligned} \hat{\mathbf{x}}_{k+1|k+1} &= \hat{\mathbf{x}}_{k+1|k} + K_{k+1}\epsilon_{k+1}, \\ P_{k+1|k+1} &= (I - K_{k+1}H_{k+1})P_{k+1|k}, \end{aligned} \tag{2.10}$$

where:

$$\begin{aligned} \epsilon_{k+1} &= y_{k+1} - H_{k+1}\hat{\mathbf{x}}_{k+1|k}, \\ S_{k+1} &= H_{k+1}P_{k+1|k}H_{k+1}^T + R_{k+1}, \\ K_{k+1} &= P_{k+1|k}H_{k+1}^T S_{k+1}^{-1}. \end{aligned} \tag{2.11}$$

The performance of the filter and the accuracy of the estimate are highly dependent on the chosen motion and measurement models and the noise covariances. Some of the common motion and measurement models used for point target tracking are described in the following sections. These models can then be expanded for multiple targets and extended targets. In this work, all the objects are assumed to move in a two-dimensional plane but they can all be extended to include three-dimensional movement.

2.1.3 Motion Models

As explained previously, motion models describe the dynamics of an object. Here, the point target state at time step k is denoted by $\mathbf{x}_k$ which can include the position $\mathbf{p} = \begin{bmatrix} p_x & p_y \end{bmatrix}^T$, the velocity $\mathbf{v} = \begin{bmatrix} v_x & v_y \end{bmatrix}^T$ and the acceleration $\mathbf{a} = \begin{bmatrix} a_x & a_y \end{bmatrix}^T$ depending on the model. The linear motion model equation with additive noise is shown in (2.6a). A motion model can also be described as a transition density $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ [21].

A common motion model is the Constant Velocity (CV) model [22]. Assuming that $\mathbf{x}_k = \begin{bmatrix} \mathbf{p}^T & \mathbf{v}^T \end{bmatrix}^T$, the discrete time state-space representation of the linear model is given in (2.6a) and the F and Q matrices are:

$$F = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.12}$$

$$Q = \sigma_Q^2 \begin{bmatrix} \frac{T_s^4}{4} & 0 & \frac{T_s^3}{2} & 0 \\ 0 & \frac{T_s^4}{4} & 0 & \frac{T_s^3}{2} \\ \frac{T_s^3}{2} & 0 & T_s^2 & 0 \\ 0 & \frac{T_s^3}{2} & 0 & T_s^2 \end{bmatrix}, \tag{2.13}$$

where T_s can be the sampling time of the sensor. The motion noise for this model will then be equal to $\mathbf{q}_k \sim \mathcal{N}(\mathbf{q}_k; \mathbf{0}_{n\times 1}, Q)$. This model is commonly used for describing vehicles with approximately constant speed, moving fairly straight, e.g., vehicles on highways.

2.1.4 Measurement Models

Measurement models relate the measurement $\mathbf{z}_k$ at time step k to the target state $\mathbf{x}_k$ and describe the likelihood of obtaining a measurement conditioned on the current state estimate, which can be represented as the density $p(\mathbf{z}_k|\mathbf{x}_k)$. In point object tracking, it is assumed that each target generates only one measurement at each time step which depends on the properties of the sensors collecting the measurements. For example, radars observe the range (distance of the object from the radar) and bearing (the angle at which the object is positioned with relation to the radar) as well as the range rate (the object velocity relative to the radar), see Figure 2.1. The range and bearing values can either be used directly in the filter which can be done with a nonlinear measurement model, or they can be transformed into a position vector $p = [p_x, p_y]^T$ and the position values can then be fed to the filter. The resulting measurement model will then be linear. The latter was chosen in this project.

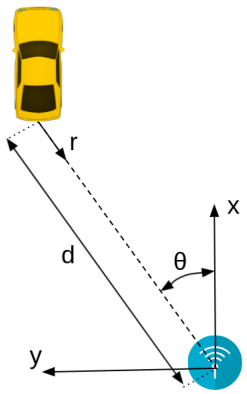


Figure 2.1: Figure depicting a schematic of a radar (blue icon) and the values measured by the radar which includes the range (d), the range rate (r) and bearing (θ).

The nonlinear Range Bearing measurement model used for radars for a single target

can be described as

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{r}_k, \tag{2.14}$$

$$h(\mathbf{x}_k) = \begin{bmatrix} d_k(\mathbf{x}_k) \\ \theta_k(\mathbf{x}_k) \end{bmatrix} = \begin{bmatrix} \sqrt{p_x^2 + p_y^2} \\ atan(\frac{p_y}{p_x}) \end{bmatrix}, \tag{2.15}$$

where $\mathbf{x}_k = [p_x, p_y, v_x, v_y]^T$ is the kinematic state at time step k. Furthermore, $d_k(\mathbf{x}_k)$ and $\theta_k(\mathbf{x}_k)$ are the range and bearing measured by the radar. The noise in this model, $\mathbf{r}_k$, is zero mean Gaussian additive noise with covariance R_k. $\mathbf{r}_k$ is dependent on the type and quality of the sensor. Note that $\mathbf{r}_k$ in these equations denotes the measurement noise and not the range rate.

Given position measurements p_x and p_y, one can use the linear measurement model in (2.16) to predict the position of the generated measurements in the update step of a Kalman filter:

$$\mathbf{z}_k = H\mathbf{x}_k + \mathbf{r}_k, \tag{2.16}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{2.17}$$

2.2 Extended Multiple Target Tracking (EMTT)

The Multiple Target Tracking (MTT) problem consists of tracking an undefined time-varying number of objects. For point target MTT, each object can be associated to at most one detection per time step and the idea is to determine the number of objects present, and the state of the dynamic targets using sensor measurements. In EMTT, the extent of the target is unknown and it should be estimated alongside the position and kinematics of the target [14]. The total state vector will then include the target position, kinematics, and extent.

In EMTT, target states and measurements can be represented by Random Finite Sets (RFS). An RFS is a random variable whose possible outcomes are sets with a finite number of unique elements. In an RFS the set size, also called the cardinality, is a random variable and the set elements are also random variables that belong to the same space (have the same dimensions). Sets are invariant to order and do not contain repeated elements, which makes them the ideal method of representing the objects and measurements in MTT.

The target states are denoted by $\mathbf{X}_k$, where k is the discrete-time step and each single target state is denoted by $\mathbf{x}_k$, where $\mathbf{x}_k \in \mathbf{X}_k$. The target set is an RFS, meaning that the cardinality $|\mathbf{X}_k|$ is a discrete random variable, and each target state $\mathbf{x}_k$ is also a random variable. The set of measurements at each time step is an RFS denoted by $\mathbf{Z}_k$, which includes both clutter (measurements associated to the

background or noise from the sensors) and target measurements. A single measurement from the set is denoted $\mathbf{z}_k$. The sequence of all measurement sets, from an initial time up to and including time step k is denoted by $\mathbf{Z}^k$ [1]. The prediction and update equations for this case are similar to the equations shown in Section 2.1.

Assuming that $f_{k|k}(\mathbf{X}_k|\mathbf{Z}^k)$ is the prior density, $f_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{X}_k)$ is the transition density and $f_{k+1}(\mathbf{Z}_{k+1}|\mathbf{X}_{k+1})$ is the measurement likelihood, the predicted density $f_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{Z}^k)$ can be calculated with the Chapman-Kolmogorov prediction:

$$f_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{Z}^k) = \int f_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{X}_k) f_{k|k}(\mathbf{X}_k|\mathbf{Z}^k)\, d\mathbf{X}_k. \tag{2.18}$$

The multi-target posterior set density $f_{k+1|k+1}(\mathbf{X}_{k+1}|\mathbf{Z}^{k+1})$ can then be computed using the Bayes' update which is described by:

$$f_{k+1|k+1}(\mathbf{X}_{k+1}|\mathbf{Z}^{k+1}) = \frac{f_{k+1}(\mathbf{Z}_{k+1}|\mathbf{X}_{k+1}) f_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{Z}^k)}{\int f_{k+1}(\mathbf{Z}_{k+1}|\mathbf{X}_{k+1}) f_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{Z}^k)\, d\mathbf{X}_{k+1}}. \tag{2.19}$$

2.2.1 Data Association

One of the challenges in multiple object tracking is determining the source of each observation. A measurement could be from an object (here called a detection) or from the background (clutter). Deciding whether a measurement is a detection or clutter is not a trivial task. Additionally, there might be several objects, meaning that one should also find out which measurement belongs to which object. Therefore, this task can be seen as a data association problem where a group of measurement values is assigned to the set of measurement sources or objects such that a certain cost criterion (e.g., distance) is minimized. The data association problem is made even more difficult in extended multiple object tracking, where each object can generate several measurements.

One approach to solving this problem is by keeping track of different hypotheses and calculating the likelihood of the occurrence of each hypothesis. In target tracking, a *local* hypothesis is a guess that a measurement or a group of measurements are generated from a certain object or clutter at a certain time step. A track is a collection of single target hypotheses corresponding to the same potential target. For each object, there can be a local hypothesis tree where each leaf of the tree is one local hypothesis for that object. For example, consider two objects being tracked and there is one measurement z_1^1 at time step 1, two measurements at time step 2 ($z_2^1,\ z_2^2$) and zero measurements for time step 3. For the sake of simplicity, the targets are assumed to be point objects. Thus, each measurement can be associated to only one object. The first object will either be detected (z_1^1 is a detection of the first object), or not detected. The same alternatives will be true for the second object, and the local hypotheses for the three time steps can be presented by the local hypotheses trees shown in Figure 2.2 where each tree corresponds to one object, and every level in the trees is one time step. In a real filtering scenario, other hypothesis trees might show up when more measurements occur, but here, we are thus not considering new possible object births.

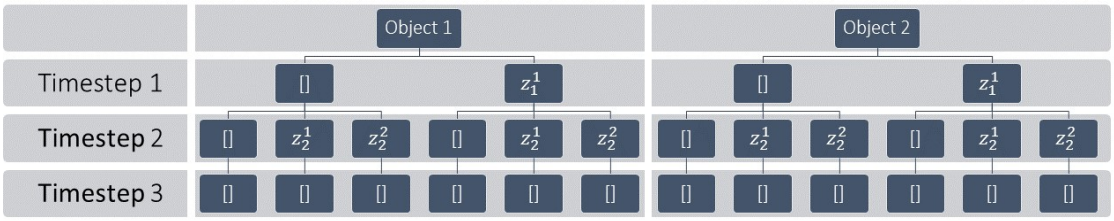


Figure 2.2: Figure showing two local hypothesis trees, one for object one and one for object two. In the leaves, [] means that no measurement point is associated to the object.

A *global* hypothesis is a collection of local hypotheses that may happen jointly. It includes local hypotheses for all considered objects, but only one local hypothesis per object. Using the leaves of the hypothesis trees, a global hypothesis lookup table (GHLT) can be created. The GHLT will include all possible combinations (global hypotheses) of assignments from the local hypothesis trees. In this case, there are three ways in total for time step 1, or three global hypotheses, describing how one can assign the measurement in the example above to either of the two targets or to clutter. The first is that both objects, let us call them o_1 and o_2, are misdetected and z_1^1 is a detection of a new object or clutter. Thus, the first local hypothesis for both objects in time step 1, shown as the left leaf for each object in Figure 2.2, is used as the first global hypothesis. The second global hypothesis is that o_1 is misdetected and that z_1^1 is assigned to o_2 The third hypothesis is that z_1^1 is assigned to o_1 and o_2 is misdetected. This is described in Table 2.1.

The lookup table for the second time step is presented in Table 2.2, where it also is illustrated which new global hypotheses are coming from which of the three global hypotheses in the previous time step. Using the local hypotheses trees again, if at the previous time step assuming both objects as misdetected, then there are seven possible options at this time step: assuming both objects are being misdetected again, assuming o_1 to be misdetected and o_2 to be detected by z_2^1 or z_2^2, o_1 being detected by z_2^1 and o_2 misdetected or detected by z_2^2 Finally, o_1 could also be detected by z_2^2 and o_2 either misdetected or detected by z_2^1. The same reasoning follows for the other global hypotheses in time step one, which causes several global hypotheses in time step 2 to be the same, see Table 2.2, but originating from different previous global hypotheses.

The third lookup table, for time step 3, has the same number of global hypotheses as time step 2, where each global hypothesis consists of misdetections for both objects since there are no measurements at this time step.

Table 2.1: Global Hypotheses Lookup Table for time step 1, where h_i is a global hypothesis and o_1 and o_2 are object 1 and object 2 respectively. The notation [] means that no measurement point is associated to the object.

$GHLT_1$			
	h_1	h_2	h_3
o_1	[]	[]	z_1^1
o_2	[]	z_1^1	[]

Table 2.2: Global Hypotheses Lookup Table for time step 2, where h_i is a global hypothesis and o_1 and o_2 are object 1 and object 2, respectively. The notation [] means that no measurement point is associated to the object.

$GHLT_2$																					
	From h_1 in $GHLT_1$							From h_2 in $GHLT_1$								From h_3 in $GHLT_1$							
	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8	h_9	h_{10}	h_{11}	h_{12}	h_{13}	h_{14}	h_{15}	h_{16}	h_{17}	h_{18}	h_{19}	h_{20}	h_{21}
o_1	[]	[]	[]	z_2^1	z_2^1	z_2^2	z_2^2	[]	[]	[]	z_2^1	z_2^1	z_2^2	z_2^2	[]	[]	[]	z_2^1	z_2^1	z_2^2	z_2^2
o_2	[]	z_2^1	z_2^2	[]	z_2^2	[]	z_2^1	[]	z_2^1	z_2^2	[]	z_2^2	[]	z_2^1	[]	z_2^1	z_2^2	[]	z_2^2	[]	z_2^1

As can be seen in the tables, the number of global hypotheses grows very quickly because, theoretically, every possible way to partition the set of measurements should be considered. In extended target tracking, the data association problem is more complicated since several measurements can be associated to one object. The solution to reduce the number of global hypotheses is to create subsets of the measurements also called clusters or partitions and then only consider how these clusters can be associated to different objects. This means that, instead of a single measurement, a set of measurements is associated to an existing or new track. Different clustering methods can be used to create the partitions. In this work, DBSCAN was used with different parameters to create an appropriate number of measurement partition sets.

2.2.2 DBSCAN

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a density-based clustering algorithm that groups together data points that are relatively close to each other, considering Euclidean distance [23]. The main idea with this method of clustering is that within the cluster, the typical density of points is higher than outside of the cluster. The clusters can vary based on the distance between the data points and the minimum number of points that should be in a cluster. If a data point is not assigned to a cluster, it is considered to be clutter.

In order to find a cluster, we start at a random point and find all of the points that are close to it based on the desired threshold and the minimum number of points. Then these new points are investigated as well, and points close to them are considered to be a part of the same cluster. This continues until there are no points left that could be within the cluster. To form the next cluster the process

is repeated with another point until all points are either within a cluster or clutter [23]. Algorithm 1 shows a simple version of the DBSCAN algorithm.

Algorithm 1 The DBSCAN Clustering Algorithm

Inputs: Set of points $P = \{p_i\}_{i=1}^{n}$, cluster size threshold ϵ and minimum number of points per cluster m.
Outputs: Cluster index set for the data points.

for $i = 1, \ldots, |P|$ **do**
    if p_i is unclassified **then**
        Find $P_i = \{p_j \,|\, p_j \in P,\ p_j \text{ is not classified},\ d(p_i, p_j) < \epsilon\}$.
        if $|P_i| < m$ **then**
            The point p_i is clutter.
            Continue to the next point in P.
        else
            P_i is a cluster.
            for each $p_j \in P_i$ **do**
                Find $P_{ij} = \{p_k \,|\, p_k \in P,\ p_k \text{ is not classified},\ d(p_j, p_k) < \epsilon\}$.
                if $|P_{ij}| \geq m$ **then**
                    for $k = 1, \ldots, |P_{ij}|$ **do**
                        if p_k is unclassified **then**
                            Add p_k to P_i and set as classified.

2.2.3 Solving the Linear Assignment Problem

Data association, which can be seen as a linear assignment problem, can be solved using several methods. The Hungarian linear assignment method and the Murty k-best assignment method are the ones used in this work. The Hungarian method is described by the following example, where a simple assignment problem can be formulated as a data-assignment problem as such:

"How should a set of measurement clusters be assigned to a set of targets such that no more than one cluster is assigned to each target and that a certain cost criterion is minimized (or gain is maximized)?"

The Hungarian algorithm, also known as the Munkres assignment algorithm, solves the optimal assignment problem using Algorithm 2 [24]. The Murty algorithm [25, 26], can be used to find the k best solutions to an assignment problem. Algorithm 3 shows the pseudo code for this method.

2.3 The Poisson Multi-Bernoulli Mixture Filter

In this study, the focus is on PMBM-based filtering methods since they have been shown to have state-of-the-art performance. The PMBM filter is based on the

Algorithm 2 The Hungarian Assignment Algorithm

1: **Inputs**: Set of n clusters ($i = 1, 2, \ldots, n$), n targets ($j = 1, 2, \ldots, n$) and cost matrix $C = c_{ij}$, which should be a square matrix. If the cost matrix is not a square matrix, add dummy rows and columns with value 0 to turn it into one.
2: **Outputs**: The optimal assignment of clusters to targets.
3:
4: For each row in the cost matrix, find the lowest value and subtract it from each element in the row.
5: For each column in cost matrix, find the lowest value and subtract it from each element in the column.
6: Mark the rows and columns that include zeros. If there are n marked lines, we've found the optimal assignment which is (i, j) such that $c_{ij} = 0$.
7: If not, find the smallest value that is not a member of any these marked rows and columns and subtract it from each element that are not included in the marked lines and add it to the elements included in the marked rows and columns.
8: Continue until the optimal assignment is found.

Algorithm 3 Murty's k Best Assignments Algorithm

1: **Inputs**: Set of n clusters ($i = 1, 2, \ldots, n$), m targets ($j = 1, 2, \ldots, m$), cost matrix $C = c_{ij}$ and the number of solutions k.
2: **Outputs**: The k optimal assignments of clusters to targets.
3:
4: Find the best assignment $a_1^* = \{(i_1, j_1), \ldots, (i_n, j_n)\}$ using an optimal assignment algorithm such as the Hungarian algorithm.
5: Denote the nodes in the best assignment by $N_1, N_2, \ldots, N_{n-1}$, where $N_1 = \{\overline{(i_1, j_1)}\}$, $N_2 = \{(i_1, j_1), \overline{(i_2, j_2)}\}$, and similarly $N_{n-1} = \{(i_1, j_1), (i_2, j_2), \ldots, \overline{(i_{n-1}, j_{n-1})}\}$. The set of all possible assignments is denoted by $\mathbb{A}$ and each node is a non-empty subset of $\mathbb{A}$. The bar above one cell in each node indicates exclusion of that cell for the next calculation. The exclusion is necessary to find if there is a next best solution regardless of how the association is made. The nodes are mutually disjoint with union $\mathbb{A} \setminus \{a_1^*\}$.
6: For a general stage k, assume that assignments $a_1^*, a_2^*, \ldots, a_{k-1}^*$ have been determined. The best assignment for node N_i and cost $C(a_i)$ is denoted by a_i. The global best assignment for stage k will then be equal to $a_k^* = a_i^*$ where $i^* = argmin_i C(a_i)$.

PMBM conjugate prior for extended target tracking. It models the targets as a union of two disjoint random finite sets of detected targets and undetected targets, which, alongside the conjugacy, allows for an appropriate Bayesian formulation of the extended target tracking problems [27].

2.3.1 PMBM Density

The set of targets $\mathbf{X}$ in the PMBM filter is defined as a union of two disjoint random finite sets representing the undetected targets $\mathbf{X}^u$ and the detected targets $\mathbf{X}^d$ which are modeled as a Poisson Point Process (PPP) and a Multi-Bernoulli Mixture (MBM) distribution, respectively.

A PPP is an RFS with a Poisson distributed set size. It has independent and identically distributed (i.i.d) elements, with an intensity of $D(\mathbf{x})$ over one target state set. The intensity can be represented as $D(\mathbf{x}) = \mu f(\mathbf{x})$ where μ is the positive Poisson rate and $f(\mathbf{x})$ is the spatial distribution. The PPP density is:

$$f(\mathbf{X}) = e^{-<D;1>} \prod_{\mathbf{x}\in\mathbf{X}} D(\mathbf{x}) = e^{-\mu} \prod_{\mathbf{x}\in\mathbf{X}} \mu f(\mathbf{x}), \tag{2.20}$$

where $\langle \cdot;\cdot \rangle$ denotes the inner product. A Bernoulli process is an RFS which has the distribution:

$$f(\mathbf{X}) = \begin{cases} 1-r, & \mathbf{X} = \emptyset \\ rf(\mathbf{x}), & \mathbf{X} = \{\mathbf{x}\} \\ 0, & |\mathbf{X}| \geq 2 \end{cases}, \tag{2.21}$$

where r is the probability of existence and $f(\mathbf{x})$ is the existence-conditioned probability density function of a single element.

The Multi-Bernoulli (MB) density for a set $\mathbf{X}$ is represented as

$$f(\mathbf{X}) = \begin{cases} \sum_{\uplus_{i\in\mathbb{I}}\mathbf{X}^i=\mathbf{X}} \prod_{i\in\mathbb{I}} f^i\mathbf{X}^i, & |\mathbf{X}| \leq |\mathbb{I}| \\ 0, & |\mathbf{X}| > |\mathbb{I}| \end{cases}, \tag{2.22}$$

where $\mathbb{I}$ is the index set of all the Bernoullis in the Multi-Bernoulli and $\mathbf{X}$ is a union of a fixed number of independent Bernoulli RFSs $\mathbf{X}^i$ where $\mathbf{X}^i \cap \mathbf{X}^j = \emptyset$, $i, j \in \mathbb{I}$ and $\mathbf{X} = \cup_{i\in\mathbb{I}}\mathbf{X}^i$. An MBM density is a normalized, weighted sum of MB densities, which can be represented by the parameters:

$$\{\Big(W^j, \{r^{j,i}, f^{j,i}(.)\}_{i\in\mathbb{I}^j}\Big)\}_{j\in\mathbb{J}}, \tag{2.23}$$

where W^j is the weight for the jth MB, $r^{j,i}$ and $f^{j,i}(.)$ are the existence probability and the existence-conditioned pdf for Bernoulli i in the jth MB, $\mathbb{J}$ is the Multi-Bernoulli index set and $\mathbb{I}^j$ is the Bernoulli index set for the jth MB.

To connect these densities to an example, assume two objects being in the field of view of a radar. The field of view (FOV) is the area that the radar can observe.

Then, each object will be represented by a Bernoulli. The MB will in this case be one global hypothesis, e.g., including both objects. An MBM includes several global hypotheses, e.g. including the previously defined MB and an MB assuming one object to be true and one to be clutter. The target set is thus a PMBM density with the parameters shown below:

$$D,\ \{\Big(W^j, \{r^{j,i}, f^{j,i}(.)\}_{i\in\mathbb{I}^j}\Big)\}_{j\in\mathbb{J}}\,, \tag{2.24}$$

where D is the PPP intensity, and W^j, $r^{j,i}$ and $f^{j,i}(\cdot)$ are the MBM parameters. When solving the MTT problem with this filter the number of Bernoulli components in the MBM increases exponentially and after only a few time steps handling all of the hypotheses becomes intractable. This means that appropriate reduction methods, such as pruning and merging, should be used to reduce the number of PPP components and Bernoullis.

PMBM densities are closed under the prediction and update steps, meaning that for standard multi-object models, choosing a PMBM prior will ensure that the predicted and posterior distributions are also PMBM without approximation. This means that the PMBM density is a conjugate prior in this context [28, 29]. The conjugacy allows us to know the theoretically exact form of the density and find approximations of the density parameters using metrics such as the Kullback Leibler Divergence (KLD), which is explained further in Section 2.4. Thus when performing the prediction and update steps in the PMBM filtering recursions, one only needs to compute the new PMBM density parameters shown in (2.24).

2.3.2 Extended Target Measurement Model

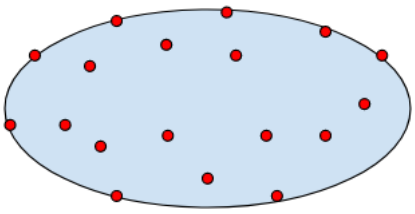


Figure 2.3: Figure depicting an ellipse-shaped object and possible measurements generated from it.

In this thesis, it is assumed that the measurements are independent and identically distributed (i.i.d). They are assumed to be generated from both the contour of and the area within the object, which is assumed to be ellipse-shaped, see Figure 2.3, where the red dots are measurements and the black contour represents the object. The PPP extended target measurement model is used to represent these measurements. The set of measurements $\mathbf{Z}_k = \{\mathbf{z}^m\}_{m\in\mathbb{M}}$ consists of clutter and target-generated measurements which are independent. Here, $\mathbb{M}$ is the measurement index

set. A target $\mathbf{x}$ is detected with probability of detection $P_D(\mathbf{x})$ and if detected, the measurements are modelled as a PPP with intensity $\gamma(x)\phi(\mathbf{z}|\mathbf{x})$, where $\gamma(\mathbf{x})$ is the measurement rate and $\phi(\mathbf{z}|\mathbf{x})$ is the spatial distribution of the measurements, which is assumed to be Gaussian in this work. The clutter is modelled as a PPP with intensity $\kappa(\mathbf{z}) = \lambda c(\mathbf{z})$, where λ is the clutter rate and $c(\mathbf{z})$ is the clutter density. The conditional measurement set likelihood for a non-empty set of measurements is described by:

$$\ell_{\mathbf{Z}}(\mathbf{x}) = P_D(\mathbf{x})p(\mathbf{Z}|\mathbf{x}) = P_D(\mathbf{x})e^{-\gamma(\mathbf{x})} \prod_{\mathbf{z}\in\mathbf{Z}} \gamma(\mathbf{x})\phi(\mathbf{z}|\mathbf{x}). \tag{2.25}$$

The probability of generating at least one detection is $1 - e^{-\gamma(\mathbf{x})}$, which means that the probability that a target is detected will be equal to $P_D(\mathbf{x})(1 - e^{-\gamma(\mathbf{x})})$. The probability that a target is not detected is equal to $q_D(\mathbf{x}) = 1 - P_D(\mathbf{x})(1 - e^{-\gamma(\mathbf{x})})$ and thus the conditional likelihood for an empty set of measurements will be equal to:

$$\ell_{\emptyset}(\mathbf{x}) = q_D(\mathbf{x}) = 1 - P_D(\mathbf{x}) + P_D(\mathbf{x})e^{-\gamma(\mathbf{x})}. \tag{2.26}$$

2.3.3 Extended Target Motion Model

In MTT, objects can appear and disappear at every time step. The appearance of a new object is called object birth and the disappearance of an object is called object death. In this work, object birth is modeled by a PPP with intensity $D_k^b(\mathbf{x})$ and at each time step N^b, new targets appear independent of existing targets according to the target birth model.

The targets survive from time step k to time step $k+1$ with the probability of survival $P_S(\mathbf{x}_k)$ and the independent target states evolve based on the transition density $f_{k+1,k}(\mathbf{x}_{k+1}|\mathbf{x}_k)$, which depends on the dynamics of the object and the spatial representation.

2.3.4 Extended Target State Density Representation

There are several models which take into account the spatial distribution of the targets as well as the kinematic state of the targets. In this work, the Random Matrix Approach was used to represent the targets.

The Gaussian inverse Wishart (GIW), which is a random matrix model, relies on the elliptic shape. It models the kinematic state as a Gaussian distribution and the spatial distribution of the object as an inverse Wishart distribution shown below:

$$\mathcal{IW}_d(\chi; v, V) = \frac{|V|^{\frac{v}{2}}}{2^{\frac{vd}{2}}\Gamma_d(\frac{v}{2})}|\chi|^{-\frac{v+d+1}{2}}e^{-\frac{1}{2}tr(V\chi^{-1})}, \tag{2.27}$$

where χ denotes the extent, v is a scalar parameter representing the degrees of freedom, d is the dimensionality of the extent, and $V \in \mathbb{R}^{d\times d}$ is the extent matrix. Both the spatial distribution of the Poisson components and the existence-conditioned

pdf of the Bernoullis are represented by Gamma Gaussian inverse Wishart (GGIW) distributions. The GGIW is based on the GIW distribution but it incorporates the estimation of the Poisson target measurement rate as well. The Gamma distribution, the pdf of which is shown below, is used to represent the density of the measurement rate γ [14]:

$$\mathcal{GAM}(\gamma; a, b) = \frac{b^a \gamma^{a-1} e^{-b\gamma}}{\Gamma(a)}, \tag{2.28}$$

where, $\Gamma(.)$ is the Gamma function [30]. The state vector in this case is comprised of:

- The kinematic state ξ, which includes the dynamic properties of the object such as position, velocity or turn-rate.
- The extent χ, which includes parameters that determine the shape and the size of the object, as well as the orientation of the shape (e.g., length and width of vehicle).
- The measurement rate γ, which determines how many measurements are generated at each time step.

The total target state density in this case can be represented by:

$$\begin{aligned} f(\mathbf{x}) &= \mathcal{GGIW}(\mathbf{x}; a, b, m, P, v, V) \\ &= \mathcal{GAM}(\gamma; a, b)\ \mathcal{N}(\xi; m, P)\ \times\ \mathcal{IW}_d(\chi; v, V), \end{aligned} \tag{2.29}$$

where, $\mathcal{GAM}(\gamma; a, b)$ is a Gamma distribution with shape a and rate b, both of which are scalar, $\mathcal{N}(\xi; m, P)$ is a Gaussian density with mean $m \in \mathbb{R}^n$ and covariance $P \in \mathbb{R}^{n \times n}$, and $\mathcal{IW}_d(\chi; v, V)$ is an inverse Wishart density with scalar degrees of freedom parameter v and parameter matrix $V \in \mathbb{R}^{d \times d}$ [31]. Further, the extent can be dependent on the kinematics, which is represented by $\times$.

The motion model used in this work for the prediction of the GGIW density consists of three models used to predict the Gaussian, the inverse Wishart, and the Gamma density parameters:

$$\xi_{k+1} = F(\xi_k) + \mathbf{q}_k\,, \tag{2.30}$$

$$\chi_{k+1} = M(\xi_k)\chi_k M(\xi)^T\,, \tag{2.31}$$

$$\gamma_{k+1} = \gamma_k\,. \tag{2.32}$$

The Gaussian kinematic state ξ is predicted using a kinematic motion model, see section 2.1.3. The prediction of extent χ which is shown in (2.31) is similar to that of the process noise covariance of a Gaussian distribution. The Transformation matrix $M(\xi)$ represents the changes in extent resulting from the movement of the object which means it is dependent on the object's kinematic state ξ. This results in nonlinearities in the model, which is why $M(\xi)$ is assumed to be equal to the identity matrix $\mathbf{I}_{d \times d}$ to obtain a linear model. This is a reasonable assumption since the object is moving with constant velocity and does not change direction so often

and the extent should remain the same over time [14]. The measurement rate γ is also assumed to be constant over time as seen in (2.32).

2.3.5 PMBM Prediction

The predicted PMBM density given a standard dynamic model, is a PMBM with parameters:

$$D_+, \{\left(W_+^j, \{r_+^{j,i}, f_+^{j,i}(.)\}_{i\in\mathbb{I}_+^j}\right)\}_{j\in\mathbb{J}_+}, \tag{2.33}$$

where the predicted Poisson density D_+ is dependent on the probability of survival, the motion model, and the birth model. Each Poisson component survives from the previous time step to the next with the probability of survival P_S, which means that the Poisson weights are multiplied by P_S. Furthermore, at each time step, there is a possibility of new potential objects appearing in the FOV. This is modeled by appending the Poisson birth model to the predicted PPP at each step and re-normalizing the Poisson weights. The predicted PPP can then be calculated as:

$$D_+ = \sum_{n=1}^{N^b} w^{b,n}\mathcal{GGIW}(\mathbf{x};\zeta^{b,n}) + \sum_{n=1}^{N^u} w^{u,n}P_S(\hat{\mathbf{x}}^{u,n})\mathcal{GGIW}(\mathbf{x};\zeta_+^{u,n}), \tag{2.34}$$

where, "u, n'' denotes the nth Poisson component, "b, n'' denotes the nth birth component, and N_u and N_b are the total number of Poisson components and birth components, respectively. In these equations and further, it is assumed that $P_S(\mathbf{x}) = P_S(\hat{\mathbf{x}})$, and $P_D(\mathbf{x}) = P_D(\hat{\mathbf{x}})$, where $\hat{\mathbf{x}}$ is the expected value of $\mathbf{x}$, i.e., $\hat{\mathbf{x}} = \int \mathbf{x}f(\mathbf{x})d\mathbf{x}$. The vector ζ_+, which is the predicted GGIW state density, is calculated by performing prediction for the GGIW with parameters $\zeta = \{a, b, m, P, v, V\}$:

$$\zeta_+ = \begin{cases} a_+ = \frac{a}{\eta} \\ b_+ = \frac{b}{\eta} \\ m_+ = Fm \\ P_+ = FPF^T + Q \\ v_+ = 2d + 2 + e^{\frac{-T_s}{\tau}}(v - 2d - 2) \\ V_+ = e^{\frac{-T_s}{\tau}} M(\xi)VM(\xi)^T \end{cases}, \tag{2.35}$$

where, d denotes the dimension of the extent, T_s is the sampling time, τ is the maneuvering correlation constant or extent decay rate which regulates the rate at which the extent grows or shrinks over time, and η is the measurement rate parameter which affects a and b. Note that the kinematic parameters m and P are predicted using the Kalman filter prediction equations.

The predicted MBM also depends on P_S such that each Bernoulli component which is a detected object survives from the previous time step to the next one with the probability of survival P_S, which allows us to model object death or disappearance from the FOV. Thus the Bernoulli existence probabilities are multiplied by P_S, the predicted Bernoulli state densities are calculated using the equations in (2.35) and the MBM weights remain the same in the prediction step, which can be described as:

$$\begin{aligned} r_+^{j,i} &= P_S(\hat{\mathbf{x}}^{u,n}) r^{j,i}, \\ f_+^{j,i} &= f^{j,i} \mathcal{GGIW}(\mathbf{x}; \zeta_+^{j,i}), \\ W_+^j &= W^j, \end{aligned} \tag{2.36}$$

where, "j, i'' denotes the ith Bernoulli in the jth multi-Bernoulli.

2.3.6 PMBM Update

The goal in the update step is to find the posterior PMBM parameters in (2.24) given the predicted parameters in (2.33) and the measurement set $\mathbf{Z}_k$. The PMBM update consists of several steps. The measurement set is first partitioned into disjoint clusters several times as explained in Section 2.2.1 to create different global hypotheses. The set $\mathbb{M}$ is the index set of the current measurement set $\mathbf{Z} = \{\mathbf{z}^m\}_{m \in \mathbb{M}}$.

Detection and misdetection hypotheses are created for each Bernoulli density (detected object). The detection hypotheses are created by updating the Bernoulli with the measurement clusters as shown below:

$$\begin{aligned} r_{\mathbf{C}}^{j,i} &= 1\,, \\ f_{\mathbf{C}}^{j,i}(\mathbf{x}) &= \mathcal{GGIW}(\mathbf{x}; \zeta_{\mathbf{C}}^{j,i})\,, \\ \mathcal{L}_{\mathbf{C}}^{j,i} &= r_+^{j,i} P_D(\hat{\mathbf{x}}_{\mathbf{C}}^{j,i}) \ell_{\mathbf{C}}^{j,i}\,. \end{aligned} \tag{2.37}$$

It is evident that if a detected object is detected again, then it must certainly exist in the field of view. Therefore when updating a Bernoulli state density with a measurement cluster $\mathbf{C}$, the existence probability is changed to 1. The likelihood for this detection hypothesis is dependent on the existence probability $r_+^{j,i}$, the probability of detection P_D and the predicted measurement likelihood for this cluster $\ell_{\mathbf{C}}^{j,i}$ which is calculated using the equation in (2.39). The GGIW state density $\zeta_{\mathbf{C}}^{j,i}$ is found by updating $\zeta_+^{j,i}$ with cluster $\mathbf{C}$. The GGIW density update is performed as shown:

$$\zeta = \begin{cases} a = a_+ + |\mathbf{C}| \\ b = b_+ + 1 \\ m = m_+ + K\epsilon \\ P = (I - KH)P_+ \\ v = v_+ + |\mathbf{C}| \\ V = V_+ + N + \hat{Z} \end{cases}, \tag{2.38}$$

And the predicted measurement likelihood for $\mathbf{C}$ is equal to:

$$\ell = \pi^{-\frac{d|\mathbf{C}|}{2}} |\mathbf{C}|^{-\frac{d}{2}} \frac{|V_+|^{\frac{v_+-d-1}{2}} \Gamma_d(\frac{v-d-1}{2}) |\hat{X}|^{\frac{|\mathbf{C}|}{2}} \Gamma(a)(b_+)^{a_+}}{|V|^{\frac{v-d-1}{2}} \Gamma_d(\frac{v_+-d-1}{2}) |\hat{R}|^{\frac{|\mathbf{C}|-1}{2}} |S|^{\frac{1}{2}} \Gamma(a_+)(b)^a}, \tag{2.39}$$

where:

$$\begin{aligned}
\bar{\mathbf{z}} &= \frac{1}{|\mathbf{C}|} \sum_{\mathbf{z}^i \in \mathbf{C}} \mathbf{z}^i, \\
Z &= \sum_{\mathbf{z}^i \in \mathbf{C}} (\mathbf{z}^i - \bar{\mathbf{z}})(\mathbf{z}^i - \bar{\mathbf{z}})^T, \\
\hat{X} &= V_+(v_+ - 2d - 2)^{-1}, \\
\epsilon &= \bar{\mathbf{Z}} - Hm_+, \\
\hat{R} &= q_{mn}\hat{X} + R, \\
S &= HP_+H^T + \frac{\hat{R}}{|\mathbf{C}|}, \\
K &= P_+H^T S^{-1}, \\
N &= \hat{X}^{\frac{1}{2}} \hat{S}^{-\frac{1}{2}} \epsilon\epsilon^T \hat{S}^{-\frac{1}{2}} \hat{X}^{\frac{T}{2}}, \\
\hat{Z} &= \hat{X}^{\frac{1}{2}} \hat{R}^{-\frac{1}{2}} Z \hat{R}^{-\frac{1}{2}} \hat{X}^{\frac{T}{2}}.
\end{aligned} \tag{2.40}$$

The parameter q_{mn} is a scaling factor, also called the multiplicative noise. Similar to the GGIW prediction, the kinematic parameters are updated using the Kalman filter update equations.

If a Bernoulli is misdetected, the existence probability $r_\emptyset^{j,i}$ decreases according to the probability of misdetection $q_D = 1 - P_D(\mathbf{x})(1 - e^{-\gamma(\mathbf{x})})$. The Bernoulli misdetection hypotheses are thus created as shown below:

$$\begin{aligned}
r_\emptyset^{j,i} &= \frac{r_+^{j,i} q_D^{j,i}}{1 - r_+^{j,i} + r_+^{j,i} q_D^{j,i}}, \\
f_\emptyset^{j,i}(\mathbf{x}) &= w_1^{j,i} \mathcal{GGIW}(\mathbf{x}; \zeta_1^{j,i}) + w_2^{j,i} \mathcal{GGIW}(\mathbf{x}; \zeta_2^{j,i}), \\
\mathcal{L}_\emptyset^{j,i} &= 1 - r_+^{j,i} + r_+^{j,i} q_D^{j,i},
\end{aligned} \tag{2.41}$$

where:

$$\begin{aligned}
q_D^{j,i} &= 1 - P_D(\hat{\mathbf{x}}^{j,i}) + P_D(\hat{\mathbf{x}}^{j,i}) \Big(\frac{b^{j,i}}{b^{j,i}+1} \Big)^{a^{j,i}}, \\
w_1^{j,i} &= (q_D^{j,i})^{-1} (1 - P_D(\hat{\mathbf{x}}^{j,i})), \\
w_2^{j,i} &= (q_D^{j,i})^{-1} P_D(\hat{\mathbf{x}}^{j,i}) \Big(\frac{b^{j,i}}{b^{j,i}+1} \Big)^{a^{j,i}}, \\
\zeta_1^{j,i} &= \zeta_+^{j,i}, \\
\zeta_2^{j,i} &= \{a_+^{j,i}, b_+^{j,i} + 1, m_+^{j,i}, P_+^{j,i}, v_+^{j,i}, V_+^{j,i}\}.
\end{aligned} \tag{2.42}$$

To create hypotheses for new detections of previously undetected objects, the PPP components are updated with the measurement clusters and transformed into Bernoullis with the parameters:

$$
\begin{aligned}
r_{\mathbf{C}}^{u} &= \begin{cases} 1 & \text{if } |\mathbf{C}| > 1 \\ \frac{\mathcal{L}_{\mathbf{C}}}{\kappa^{\mathbf{C}}+\mathcal{L}_{\mathbf{C}}} & \text{if } |\mathbf{C}| = 1 \end{cases}, \\
f_{\mathbf{C}}^{u}(\mathbf{x}) &= \frac{\sum_{n=1}^{N^u} w^{u,n} P_D(\hat{\mathbf{x}}^{u,n}) \ell_{\mathbf{C}}^{u,n} \mathcal{GGIW}(\mathbf{x}; \zeta_{\mathbf{C}}^{u,n})}{\sum_{n=1}^{N^u} w^{u,n} P_D(\hat{\mathbf{x}}^{u,n}) \ell_{\mathbf{C}}^{u,n}}, \\
\mathcal{L}_{\mathbf{C}}^{u} &= \sum_{n=1}^{N^u} w^{u,n} P_D(\hat{\mathbf{x}}^{u,n}) \ell_{\mathbf{C}}^{u,n}.
\end{aligned}
\tag{2.43}
$$

If the cluster $\mathbf{C}$ contains only one measurement, the existence probability $r_{\mathbf{C}}^{u}$ will depend on the clutter intensity κ and the likelihood $\mathcal{L}_{\mathbf{C}}^{u}$.

Once the local hypotheses are created by updating the PPP and Bernoullis with the measurement clusters, the most likely association of clusters to Bernoullis should be found. The set $\mathcal{A}$ is the set of all data associations A given the predicted hypothesis. Each cluster is assigned to a new track or an existing track. Each cluster can be associated to only one track, new or existing (predicted) and each predicted track can be assigned to either one cluster or no clusters. Clusters not assigned to a predicted track represent a new track.

Each of these clustering and cluster-track association pairs is a partition in $\mathbb{M} \uplus \mathbb{I}$, where $\mathbb{I}$ is the index set of the predicted Bernoullis. The data association A consists of partitioning this set into non empty disjoint subsets $\mathbb{C} \in A$ which are called index cells. Each index cell can have either zero or only one Bernoulli index $i^{\mathbb{C}}$ and $\mathbf{C}_{\mathbb{C}}$ is the measurement cluster corresponding to the index set $\mathbb{C}$ or $\mathbf{C}_{\mathbb{C}} = \cup_{m \in \mathbb{M} \cap \mathbb{C}} \mathbf{z}^m$.

For each global hypothesis, a cost matrix is created using the measurement likelihood values and index cells for each local hypothesis. Each row represents a measurement cluster and each column represents a track (already existing or newly created). This cost matrix is used to assign the clusters to the tracks based on the likelihood values. The best M global hypotheses or data associations are then found by using a k best assignment algorithm such as the Murty algorithm. For a data association A, the likelihood of association L_A:

$$
\begin{aligned}
L_A &= \prod_{\substack{\mathbb{C} \in A: \\ \mathbb{C} \cap \mathbb{I} = \emptyset \\ \mathbb{C} \cap \mathbb{M} \neq \emptyset}} L_{\mathbf{C}_{\mathbb{C}}}^{b} \prod_{\substack{\mathbb{C} \in A: \\ \mathbb{C} \cap \mathbb{I} \neq \emptyset \\ \mathbb{C} \cap \mathbb{M} \neq \emptyset}} L_{\mathbf{C}_{\mathbb{C}}}^{i_{\mathbb{C}}} \prod_{\substack{\mathbb{C} \in A: \\ \mathbb{C} \cap \mathbb{I} \neq \emptyset \\ \mathbb{C} \cap \mathbb{M} = \emptyset}} L_{\emptyset}^{i_{\mathbb{C}}}, \\
L_{\mathbf{C}_{\mathbb{C}}}^{b} &= \begin{cases} \kappa^{\mathbf{C}_{\mathbb{C}}} + \langle D^u; \ell_{\mathbf{C}_{\mathbb{C}}} \rangle, & if\ |\mathbf{C}_{\mathbb{C}}| = 1 \\ \langle D^u; \ell_{\mathbf{C}_{\mathbb{C}}} \rangle, & if\ |\mathbf{C}_{\mathbb{C}}| > 1 \end{cases}, \\
L_{\mathbf{C}_{\mathbb{C}}}^{i_{\mathbb{C}}} &= r^i \langle f^i; \ell_{\mathbf{C}_{\mathbb{C}}} \rangle, \\
L_{\emptyset}^{i_{\mathbb{C}}} &= 1 - r^i + r^i \langle f^i; q^D \rangle,
\end{aligned}
\tag{2.44}
$$

can be described as the product of three parts:

- measurements associated to background (clutter or previously undetected objects),
- measurements associated to a previously detected targets ,
- previously detected targets that are misdetected.

The weights of the global hypotheses, or the weights for the resulting Multi-Bernoulli, can then be calculated by normalizing the association likelihood:

$$W_A = \frac{L_A}{\sum_{A \in \mathcal{A}} L_A}. \tag{2.45}$$

Enumerating all possible associations after several time steps is computationally intractable, so it is better to truncate associations with negligible weights [14]. The PPP intensity is then updated with misdetections. The resulting PPP intensity is as described as:

$$D^u(\mathbf{x}) = \sum_{n=1}^{N^u} \Big(w_1^{u,n} \mathcal{GGIW}(\mathbf{x}; \zeta_1^{u,n}) + w_2^{u,n} \mathcal{GGIW}(\mathbf{x}; \zeta_2^{u,n}) \Big), \tag{2.46}$$

where:

$$\begin{aligned} w_1^{u,n} &= (1 - P_D(\hat{\mathbf{x}}^{u,n}))w^{u,n}, \\ w_2^{u,n} &= P_D(\hat{\mathbf{x}}^{u,n}) \Big(\frac{b^{u,n}}{b^{u,n}+1} \Big)^{a^{u,n}} w^{u,n}, \\ \zeta_1^{u,n} &= \zeta_+^{u,n}, \\ \zeta_2^{u,n} &= \{a_+^{u,n}, b_+^{u,n}+1, m_+^{u,n}, P_+^{u,n}, v_+^{u,n}, V_+^{u,n}\}. \end{aligned} \tag{2.47}$$

The global hypothesis look-up table is then updated based on the new associations. The required reduction steps (pruning, capping, merging, etc.) are then performed to reduce the number of local and global hypotheses and thus the complexity. These steps are further discussed in section 2.3.7. After reduction, the global hypothesis look-up table is re-indexed and the posterior PMBM will have the density:

$$\begin{aligned} f(\mathbf{X}|\mathbf{Z}) &= \sum_{\mathbf{X}^u \uplus \mathbf{X}^d = \mathbf{X}} f^u(\mathbf{X}^u) \sum_{j \in \mathbb{J}_+} \sum_{A \in \mathcal{A}^j} W_A^j f_A^j(\mathbf{X}^d), \\ f^u(\mathbf{X}^u) &= e^{-\langle D^u; 1 \rangle} \prod_{\mathbf{x} \in \mathbf{X}^u} D^u(\mathbf{X}), \\ f_A^j(\mathbf{X}^d) &= \sum_{\uplus_{C \in A} \mathbf{X}^C = \mathbf{X}} \prod_{C \in A} f_C^j(\mathbf{X}^C). \end{aligned} \tag{2.48}$$

2.3.7 Complexity Reduction

The computational complexity of the PMBM tends to increase over time, and if nothing is done to decrease the complexity, it quickly becomes very expensive or even unfeasible to solve [27]. Several methods to decrease the complexity are therefore used, where gating, pruning, capping, merging and recycling are some of the most common methods [32].

Gating

Gating consists of only updating Bernoullis and Poisson state densities with measurements that lay within an area around the densities. A common method of gating is ellipsoidal gating, where the gating area around the state density is formed as an ellipse. The gate has to be sufficiently wide so as not to miss the object considering the measurement noise, the extent and the movement of the object [33].

Pruning

After updating the PMBM one can reduce the number of MBs by pruning or removing MBs with weights smaller than a certain threshold and normalizing the mixture weights again. Pruning can also be done for Bernoullis within each MB by removing Bernoullis with small existence probabilities. PPP components with negligible weights can be pruned as well [29]. Note that the PPP weights should also be normalized after pruning.

Capping

In capping, the number of Bernoullis and MBs is limited to a constant value. The MBs with the highest weights and the Bernoullis with the largest existence probabilities should be chosen. This value should be tuned based on the complexity of the scenario [32].

Merging

Merging consists of combining two or more densities with similar distributions to obtain one. The merging method and its complexity depend on the distribution. For example, Gaussian densities can be merged by performing moment-matching to find the parameters of the final merged density.

Recycling

Recycling is a method used to reduce the number of Bernoullis in a Multi-Bernoulli by converting all Bernoullis with a lower existence probability r^i than a threshold τ_r to Poisson Point Process components and assuming that they are undetected objects [1]. The new PPP intensity ($D^u_{new}(x)$) is described by:

$$D^u_{new}(x) = D^u(x) + \sum_{\iota \in \hat{\mathbb{I}}: \hat{r}^\iota < \tau_r} \hat{r}^\iota g^\iota(x), \tag{2.49}$$

where, $g^i(x)$ is the existence conditioned probability density function for Bernoulli i in the approximating MB $g(x)$, thus one of the Bernoullis in the approximated MB. $\hat{\mathbb{I}}$ is the index set of the approximated Bernoullis.

2.3.8 PMBM Estimation

An optimal estimation for a target tracker uses an evaluation metric to continuously evaluate and minimize the metric. However, ground truth may not always be

available during tracking. Therefore, several sub-optimal estimators that are easy to compute have been developed for the PMBM filter. Three estimation methods, one used in MHT, one in δ-GLMB, and one developed for the PMBM are all presented and evaluated for the PMBM filter in [6] for different clutter rates and probabilities of detections. For cases similar to ours regarding clutter rate and probability of detection, the estimator developed for the PMBM yielded the best result. This method can be described in two steps, first, the global hypothesis with the highest weight is chosen according to

$$j^* = \arg\max_j W^j \,, \tag{2.50}$$

where W^j are the MBM weights and n is the number of global hypotheses. Then, all Bernoulli components with greater existence probabilities than a threshold τ_{est} are taken as estimates [6]. The threshold τ_{est} is thus representing the number of misdetections an object can have since the probability of existence depends on the probability of survival and probability of detection.

2.4 The Poisson Multi-Bernoulli Filter

As mentioned in the previous sections, when using the PMBM filter the number of Multi-Bernoullis increases exponentially, which is why in [1] the extended target PMBM prior density is estimated as an extended target PMB density by approximating the MBM in the posterior as a single Multi-Bernoulli. This was shown to lead to a lower computational cost while maintaining relatively good performance and low estimation error.

The PMB set density can be defined entirely by the parameters:

$$D^u, \{r^i, f^i(\cdot)\}_{i\in\mathbb{I}}, \tag{2.51}$$

where D^u is the PPP intensity, r^i is the existence probability of Bernoulli i, $f^i(\cdot)$ is the existence-conditioned pdf of Bernoulli i, and $\mathbb{I}$ is the Bernoulli index set. Given a prior which is a PMB, the predicted density will also be a PMB but the posterior will be a PMBM density $f_{k|k}(\cdot)$ with parameters in (2.24) due to the unknown data associations. This PMBM density should be approximated as a PMB $\hat{f}_{k|k}(\cdot)$ while retaining as much information as possible. This can be done by minimizing the Kullback-Leibler divergence (KLD) between the densities, which is a measure of distortion caused to an entire multi-target density when approximating it by another multi-target density [7]. For two densities $f_1(\mathbf{X})$ and $f_2(\mathbf{X})$, the KLD is defined as [34]:

$$D_{KL}\Big(f_1(\mathbf{X})||f_2(\mathbf{X})\Big) = \int f_1(\mathbf{X}) \log\left(\frac{f_1(\mathbf{X})}{f_2(\mathbf{X})}\right) d\mathbf{X}, \tag{2.52}$$

where, for $\mathbf{X} = \{\mathbf{x}_1, \dots \mathbf{x}_n\}$ the set integral is described by:

$$\int f(\mathbf{X})\delta\mathbf{X} = f(\emptyset) + \sum_{n=1}^{\infty} \frac{1}{n!} \int \cdots \int f(\{\mathbf{x}_1, \dots \mathbf{x}_n\}) d\mathbf{x}_1 \dots d\mathbf{x}_n. \tag{2.53}$$

The approximation problem can then be written as:

$$\hat{f}_{k|k}(\boldsymbol{X}_k) = \min_{\hat{f}} D_{KL}\Big(f_{k|k}(\boldsymbol{X}_k)||\hat{f}_{k|k}(\boldsymbol{X}_k)\Big). \tag{2.54}$$

Solving this problem analytically is intractable for EMTT and an appropriate approximation method should be used [1]. The PPP that represents the undetected target does not have to be approximated, so the main problem is approximating an MBM as an MB by minimizing the KL divergence. Assuming that $f(\mathbf{X})$ denotes the MBM, where each MB is represented by $f^j(\mathbf{X})$ and $g(\mathbf{X})$ is the approximating MB, the main problem can then be written as:

$$g(\boldsymbol{X}) = \min_{g} D_{KL}\Big(f(\boldsymbol{X})||g(\boldsymbol{X})\Big). \tag{2.55}$$

There are several challenges that one faces when solving this problem. The first is determining the number of Bernoullis in the Multi-Bernoulli RFS since the MBs at different time steps do not necessarily have the same number of Bernoullis. The second challenge is determining which Bernoullis should be merged to form the MB posterior and the last one is choosing a method for merging the Bernoullis. We can either use the merging technique in the Track-Oriented Poisson Multi-Bernoulli filter (TO-PMB) or the one used in the Variational Multi-Bernoulli (VMB) filter.

2.4.1 Track-Oriented Poisson Multi-Bernoulli

The Track-Oriented PMB (TO-PMB) filter is based on merging hypotheses belonging to the same track across different global hypotheses. An example showing how this is done is presented in Section 2.4.2. This is done to obtain a single Bernoulli representing that track. In this method, the single target hypotheses that are updating different tracks are assumed to be independent. This assumption does not always hold, since there could always exist measurement partitions that update different tracks but share measurements, and assuming that they are independent can lead to large errors. Therefore the VMB filter is preferred since it breaks down the problem into two sub-problems concerning the formation of existing tracks and the formation of new tracks. This way we can find a minimum upper bound for the KL divergence problem and thus the approximating MB.

2.4.2 Variational Multi-Bernoulli

The point target Variational Multi-Bernoulli (VMB) can be used to find the approximating MB because, in the extended target PMB filter, the number of existing tracks remains the same [1]. This means that when the PMB density is updated, the same number of predicted Bernoullis are updated in each global hypothesis, so the number of existing objects (tracks) is the same across global hypotheses. In VMB

we minimize the KLD:

$$\begin{aligned}
&\arg\min_g D_{KL}\Big(f(\boldsymbol{X})||g(\boldsymbol{X})\Big) = \arg\min_g\Big[\int f(\boldsymbol{X})\log\Big(\frac{f(\boldsymbol{X})}{g(\boldsymbol{X})}\Big)d\boldsymbol{X}\Big] \\
&= \arg\min_g\Big[\int f(\boldsymbol{X})\log(f(\boldsymbol{X}))d\boldsymbol{X}\Big] - \arg\min_g\Big[\int f(\boldsymbol{X})\log(g(\boldsymbol{X}))d\boldsymbol{X}\Big] \\
&= \arg\max_g\Big[\int f(\boldsymbol{X})\log(g(\boldsymbol{X}))d\boldsymbol{X}\Big],
\end{aligned} \tag{2.56}$$

where $f(\mathbf{X})$ is the MBM and $g(\mathbf{X})$ is the approximated MB. In the equation, the term $\int f(\boldsymbol{X})\log(f(\boldsymbol{X}))\delta\boldsymbol{X}$ is constant with respect to the minimization variables, which means that it can be neglected. This minimization problem can be solved by using the Expectation-Maximization method to get an approximate solution in form of a minimum upper bound. This method provides a mechanism for machine learning inference (deployment of a trained machine learning model in a production environment) on distributions where the machine learning process would be easier if additional data were included alongside the observations. Thus to approximate the density $u(y)$ with the density $v(y)$, we should find the $v(y, y^*)$ that maximizes the log likelihood:

$$\mathcal{L}(v) = \int u(x)\log(\sum_{y^*} v(y, y^*))dx \tag{2.57}$$

where y^* is the unobserved (latent) data or the missing data. This problem can then be solved in two steps:

- E-Step: Calculate the expectation of the missing data distribution.

$$q(y^*|y) = \frac{v(y, y^*)}{\sum_{y^*} v(y, y^*)} \tag{2.58}$$

- M-Step: Calculate the parameters of $v(y, y^*)$ which maximizes the completed log likelihood:

$$\sum_{y^*}\int q(y^*|y)u(y)\ \log(v(y, y^*))dy \tag{2.59}$$

.

Thus, the upper bound of the negative log likelihood is minimized by using a coordinate descent method. The minimization is done with respect to both the model parameters and the missing data distribution and is guaranteed to converge to a local minimum.

Formation of Existing Tracks

As discussed in the previous section, correspondences between $f(\mathbf{X})$ and $g(\mathbf{X})$ can be treated as a missing data problem. The MB integrals can be simplified to a series of Bernoulli integrals and the log-sum inequality can be used to find an approximate upper bound for the objective [35]:

$$D_{UB} = -\sum_{j\in\mathbb{J},\pi\in\Pi_N^j} W^j q^j(\pi^j)\sum_{\iota\in\mathbb{I}}\int f^{j,\pi^j(\theta(\iota))}(\mathbf{X})\ \log(g^\iota(\mathbf{X}))d\mathbf{X}, \tag{2.60}$$

where, $N = |\mathbb{I}|$ is the number of Bernoullis that correspond to existing tracks in each MB of $f(\mathbf{X})$ (which is equal for all MBs) and the approximating MB $g(\mathbf{X})$, Π_N^j is the set of all the ways to assign Bernoullis $f^{j,i}(\mathbf{X}^i)$ ($i \in \mathbb{I}$, $j \in \mathbb{J}$) to the Bernoullis $g^\iota(\mathbf{X}^\iota)$ ($\iota \in \mathbb{I}$), $q^j(\pi^j)$ is constrained to vary with the jth MB and follows $q^j(\pi^j) \geq 0$ and $\sum_{\pi^j \in \Pi_N^j} q^j(\pi^j) = 1$. This problem can be solved using block coordinate descent which should be done with approximations due to combinatorial complexity. There are two methods for solving this which result in two variations of the VMB: The Most Likely Assignment PMB (MLA-PMB) and the Efficient Approximation of a Feasible Set PMB (EAFS-PMB).

Most Likely Assignment

The Most Likely Assignment (MLA) PMB is a variation of the PMB filter which differs based on how the existing tracks are formed. The filter uses a method similar to the Kullback-Leibler Set Joint Probabilistic Data Association (KLSJPDA) filter [36]. We use the fact that in the PMBM conjugate prior, each global hypothesis (MB) has the same number of existing tracks (Bernoullis representing existing tracks) and that the multi-target RFS is order-invariant. We assume that different Bernoullis in different MBs are merged if they have the same superscript i (if they correspond to single target hypotheses updating the same track) and only Bernoullis with the same superscript can merge. Then we can permute (reorder) the Bernoullis in each MB such that the upper bound is minimized but the density of the MBM is not changed. This is done by finding the single most likely assignment $\hat{\pi}^j$ for each MB $f^j(\mathbf{X})$. Thus the missing data distribution for each MB will be a point mass $q^j(\hat{\pi}^j) = 1$.

$$\hat{\pi}^j = \arg\min_{\pi^j} -\sum_{\iota \in \mathbb{I}} \int f^{j,\pi^j(\theta(\iota))}(\mathbf{X}) \log(g^\iota(\mathbf{X}))\delta\mathbf{X}\,,\ j \in \mathbb{J} \tag{2.61}$$

This assignment problem can be solved for π^j using the Hungarian algorithm. The minimization with respect to the approximating MB $g(\mathbf{X})$ is then simplified to:

$$\arg\min_{g^\iota} D_{KL}\Big(\sum_{j \in \mathbb{J}} W^j f^{j,\hat{\pi}^j(\theta(\iota))}(\mathbf{X}^{\hat{\pi}^j(\theta(\iota))}) || g^\iota(\mathbf{X}^\iota) \Big) \tag{2.62}$$

Efficient Approximation of Feasible Set

Efficient Approximation of Feasible Set (EAFS) PMB is, similarly to the MLA-PBM, a filter type which differs from the other PMB filters regarding how the existing tracks are formed.

In this filter, since the missing data problem involves missing data for each MB in the PMBM density, the problem can be approximated as:

$$\arg\min_{q(h,\iota) \in \mathcal{M}} -\sum_{\iota \in \mathbb{I}} \int \Big(\sum_{h \in \mathbb{H}} q(h,\iota) f^h(\mathbf{X}) \Big) \log(g^\iota(\mathbf{X}))\delta\mathbf{X}, \tag{2.63}$$

where $\mathbb{H}$ is the index set of single target hypotheses included in the global hypotheses, $q(h,\iota)$ is the weight of Bernoulli $f^h(\mathbf{X}^h)$ in $f(\mathbf{X})$ that is assigned to the

approximating Bernoulli $g^\iota(\mathbf{X}^\iota)$ and $\mathcal{M}$ is the feasible set (an approximation which is needed for tractability) and it's defined as:

$$\mathcal{M} = \{q(h,\iota) \geq 0 | \sum_{h\in\mathbb{H}} q(h,\iota) = 1 \ \forall \iota \in \mathbb{I}\,,\ \sum_{\iota\in\mathbb{I}} q(h,\iota) = p_h \ \forall\, h \in \mathbb{H}\}, \tag{2.64}$$

where:

$$p_h = \sum_{\iota\in\mathbb{I}} p_\iota(h)\,, \quad p_\iota(h) = \sum_{j\in\mathbb{J}} W^j \delta_{f^{j,\theta(\iota)}(\mathbf{X}^{\theta(\iota)})}(f^h(\mathbf{X}^h))\,,\ \iota \in \mathbb{I}. \tag{2.65}$$

Here each approximating Bernoulli is formed as a weighted sum of different single target hypothesis densities. In this case the M-step will be:

$$\arg\min_{g^\iota} D_{KL}\Big(\sum_{h\in\mathbb{H}} q(h,\iota) f^h(\mathbf{X}^h) || g^\iota(\mathbf{X}^\iota)\Big) \tag{2.66}$$

and the E-step becomes:

$$\begin{aligned} &\arg\min_{q(h,\iota)} \sum_{h\in\mathbb{H}}\sum_{\iota\in\mathbb{I}} -q(h,\iota)\int f^h(\mathbf{X})\,\log\, g^\iota(\mathbf{X})\delta\mathbf{X}\,,\\ &s.t. \quad \sum_{\iota\in\mathbb{I}} q(h,\iota) = p_h \ \forall\, h \in \mathbb{H}\,,\\ &\qquad \sum_{h\in\mathbb{H}} q(h,\iota) = 1 \ \forall\, \iota \in \mathbb{I}\,,\\ &\qquad q(h,\iota) \geq 0 \ \forall\, h \in \mathbb{H}, \iota \in \mathbb{I}\,, \end{aligned} \tag{2.67}$$

which can be solved using the Simplex algorithm [37].

Formation of New Tracks

New tracks are created by merging the Bernoullis that are highly dependent across different Multi-Bernoullis. This ensures that there won't be several Bernoullis in the same region and that significantly different Bernoullis won't be merged. A suitable metric for similarity could be the Symmetrical Kullback-Leibler Divergence (SKLD) which is shown below:

$$D_{SKL}(p_1||p_2) = D_{KL}(p_1||p_2) + D_{KL}(p_2||p_1), \tag{2.68}$$

where the KL divergence between two GGIW densities is calculated as:

$$D_{KL}\Big(p_1(\mathbf{x})||p_2(\mathbf{x})\Big) = \int p_1(\mathbf{x})\,\log\left(\frac{p_1(\mathbf{x})}{p_2(\mathbf{x})}\right) d\mathbf{x}. \tag{2.69}$$

In order to calculate this value, the densities can be factorized into their three parts and the KL divergence can be computed for each part separately:

$$\begin{aligned} D_{KL}\Big(p_1(\mathbf{x})||p_2(\mathbf{x})\Big) = &D_{KL}\Big(\mathcal{GAM}(\gamma; a_1, b_1)||\mathcal{GAM}(\gamma; a_2, b_2)\Big)\\ &+ D_{KL}\Big(\mathcal{N}(\xi; m_1, P_1)||\mathcal{N}(\xi; m_2, P_2)\Big)\\ &+ D_{KL}\Big(\mathcal{IW}(\chi; v_1, V_1)||\mathcal{IW}(\chi; v_2, V_2)\Big) \end{aligned} \tag{2.70}$$

Equations (2.71), (2.72) and (2.73) show the KL divergence for the Gamma, Gaussian and the inverse Wishart distributions, respectively [38, 39].

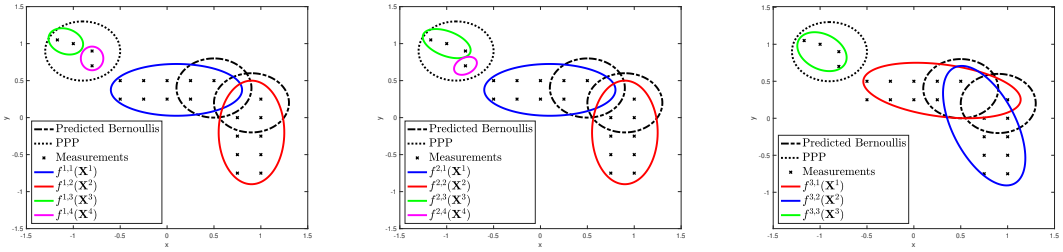
$$\begin{aligned} D_{KL}\Big(\mathcal{GAM}(\gamma; a_1, b_1)||\mathcal{GAM}(\gamma; a_2, b_2)\Big) &= a_1\ \log(b_1) - a_2\ \log(b_2) \\ &+ \log\left(\frac{\Gamma(a_2)}{\Gamma(a_1)}\right) + a_1\left(\frac{b_2}{b_1} - 1\right) \\ &+ (a_1 - a_2)\Big(\psi_0(a_1) - \log(b_1)\Big) \end{aligned} \tag{2.71}$$

$$\begin{aligned} D_{KL}\Big(\mathcal{N}(\xi; m_1, P_1)||\mathcal{N}(\xi; m_2, P_2)\Big) &= \frac{1}{2}\Big(\log|P_2| - \log|P_1| - n_x \\ &+ Tr[P_2^{-1}P_1] + (m_1 - m_2)^T P_2^{-1}(m_1 - m_2)\Big) \end{aligned} \tag{2.72}$$

$$\begin{aligned} D_{KL}\Big(\mathcal{IW}(\chi; v_1, V_1)||\mathcal{IW}(\chi; v_2, V_2)\Big) &= -\frac{v_1 - d - 1}{2}\log|V_1^{-1}V_2| \\ &+ \frac{v_2 - d - 1}{2}Tr[V_1^{-1}V_2] + \log\frac{\Gamma_d\Big(\frac{v_1 - d - j}{2}\Big)}{\Gamma_d\Big(\frac{v_2 - d - j}{2}\Big)} \\ &+ \frac{v_2 - v_1}{2}\sum_{j=1}^{d}\psi_0\Big(\frac{v_1 - d - j}{2}\Big) + d\Big(\frac{v_2 - d - j}{2}\Big) \end{aligned} \tag{2.73}$$

where Γ is the Gamma function, Γ_d is the multivariate gamma function for d which is the dimension, ψ_0 is the Digamma function and $Tr[V]$ is the trace of matrix V. n_x is the number of elements in the kinematic state vector m.

In order to further explain the differences between the TO-PMB, the MLA-PMB, and the EAFS-PMB, a simple example is provided here which is borrowed from [1]. Figure 2.4 shows an MBM which is the result of updating a predicted MB with measurements, and this MBM has three global hypotheses with two predicted Bernoullis (existing tracks) and several objects that are detected for the first time (new tracks). These new objects are rather closely spaced, so it is difficult to tell how many new objects are there. For simplicity, it is assumed that each Bernoulli has an existence probability of 1 and that there is no misdetection. All of the updated and new Bernoullis should be merged appropriately to find the approximating MB $g(\mathbf{X})$.



(a) Global Hypothesis 1 **(b)** Global Hypothesis 2 **(c)** Global Hypothesis 3

Figure 2.4: Examples of three global hypotheses, where the colored ellipses are local hypotheses. The red and the blue ellipses are representing existing tracks, while the green and magenta ones represent new tracks. The figure is taken from [1] with permission.

In TO-PMB, all local hypotheses belonging to one track are merged based on their MBM weights. Thus, the order of the Bernoullis in the MBM is important. The Bernoullis resulting from merging the existing tracks and new tracks will thus contain the Bernoullis:

$$\begin{aligned} g^1 &= W^1 f^{1,1} + W^2 f^{2,1} + W^3 f^{3,1} \\ g^2 &= W^1 f^{1,2} + W^2 f^{2,2} + W^3 f^{3,2} \\ g^3 &= W^1 f^{1,3} + W^2 f^{2,3} + W^3 f^{3,3} \\ g^4 &= W^1 f^{1,4} + W^2 f^{2,4} \end{aligned} \tag{2.74}$$

where g^ι are Bernoullis in the approximating MB, W^j are the MBM weights, and the plus sign represents the merging of densities and not a simple addition operation. We now have an approximate MB with four Bernoullis which will be used in the MLA-PMB and EAFS-PMB approximations.

In MLA-PMB, we should find the most likely permutation of Bernoulli components for each MB. In this example, the most likely permutation is found by switching the order of $f^{3,1}$ and $f^{3,2}$ because it minimizes the KLD. This means that according to this merging method, the blue Bernoullis in all global hypotheses and the red Bernoullis in all global hypotheses, belong to the same object, respectively. After the permutation, the first two Bernoulli components are defined as:

$$\begin{aligned} g^1 &= W^1 f^{1,1} + W^2 f^{2,1} + W^3 f^{3,2} \\ g^2 &= W^1 f^{1,2} + W^2 f^{2,2} + W^3 f^{3,1} \end{aligned} \tag{2.75}$$

The EAFS-PMB also uses the approximating MB found by using the TO-PMB merging method as a starting point for minimizing the KLD. In this method a portion of the weight assigning $f^{3,1}$ to g^1 should be shifted to g^2 and similarly a portion of the weight assigning $f^{3,2}$ to g^2 should be shifted to g^1. Thus, the first two

Bernoulli components can be defined as:

$$\begin{aligned} g^1 &= W^1 f^{1,1} + W^2 f^{2,1} + (W^3 - \Delta W) f^{3,1} + \Delta W f^{3,2} \\ g^2 &= W^1 f^{1,2} + W^2 f^{2,2} + (W^3 - \Delta W) f^{3,2} + \Delta W f^{3,1} \end{aligned} \tag{2.76}$$

where, both the weight of $f^{3,2}$ in g^1 and the weight of $f^{3,1}$ in g^2 are $0 \leq \Delta W \leq W^3$, given by the optimal $q(h, \iota)$.

2.5 Evaluation Metric

For point objects, the estimation results can be evaluated using the Euclidean distance as an estimation error between the ground truth and the estimated state. Further, the relation between correctly assigned elements needs to be compared to false and missed detections. This can be done using the Generalized Optimal Sub-Pattern Assignment (GOSPA) metric [40]. However, when doing extended object tracking, one needs to also consider the shape of the object. This can be done using GOSPA with the Gaussian Wasserstein (GW) distance metric, where the distance between the true and the estimated densities are measured. According to [14], it can preferably be calculated using the Gaussian-Wasserstein distance for elliptical-shaped objects.

2.5.1 Generalized Optimal Sub-Pattern Assignment Metric (GOSPA)

GOSPA [40] is an evaluation metric commonly used to evaluate target tracking performance. More specifically is it used to evaluate the estimation results compared to ground truth by including three different criteria in its evaluation; localization error, misdetections, and false detections. GOSPA is originated from the OSPA metric [41] and can be split into two parts:

1. Find an optimal one-to-one assignment between the estimated objects and the ground truth objects such that pairs are left unassigned if the distance $d(x, y)$ is larger than a comparison threshold c, i.e., $d(x, y) > c$, where x is a ground truth state and y is an estimated state. Unassigned true elements are referred to as missed detections and unassigned estimates are referred to as false detections.
2. Set the cost for the assigned and unassigned elements. Set the cost for the assigned pairs to their distance and set the cost for the unassigned elements to c/α.

The formal definition of GOSPA, in the case where the cardinality of true states $|X|$ is smaller than the cardinality of the estimated states $|Y|$ is described by

$$\left[\min_{\pi \in \Pi_{|Y|}} \left(\sum_{(i=1)}^{|X|} \left(d^{(c)}(x_i, y_{\pi(i)})^p \right) + \frac{c^p}{\alpha} \Big(|Y| - |X| \Big) \right) \right]^{\frac{1}{p}}, \tag{2.77}$$

where $\pi \in \Pi_n$ is a sequence $(\pi(1), ..., \pi(n))$ of all permutations $\{1, ..., n\}$, $n \in \mathbb{N}$. X and Y are the ground truth and the estimations, respectively, where x_i, y_i are

representing single targets in the sets. p is a penalty $1 \leq p < \infty$ defining how much the outliers are penalized and α is a parameter used to determine the cardinality mismatch together with the comparison constant c, $c > 0$ [40].

In [40], it is argued that $\alpha = 2$ is the most appropriate value to use in GOSPA. That makes the cost for a single unassigned target the same regardless of whether it is associated to another target or not. Using $\alpha = 2$, the equation can be written as

$$\left[\min_{\phi \in \Phi} \left(\sum_{(i,j) \in \phi} \left(d(x_i, y_i)^p \right) + \frac{c^p}{2} \Big(\underbrace{|X| - |\phi|}_{\#missed} + \underbrace{|Y| - |\phi|}_{\#false} \Big) \right) \right]^{\frac{1}{p}}, \tag{2.78}$$

where $|\phi|$ is the cardinality of the properly associated targets and Φ is the set of all possible assignments. In the summation, the first term represents the estimation error and the second term is a penalty for unassigned targets which consist of missed detections and false detections [40].

2.5.2 Gaussian-Wasserstein Distance

The GOSPA metric is useful to measure distances between sets, but it assumes that there is a distance metric $d(\cdot)$ that can take two state vectors and measure the distances between them. For extended objects, the distance between the states should reflect how accurately the estimated shape corresponds to the true shape, which means that standard norms such as the Euclidean norm cannot be used, which is why the Gaussian-Wasserstein distance is a good alternative.

The Gaussian-Wasserstein distance is a way of calculating the distance between elliptically shaped densities. This metric is suitable for EMTT because it considers the extent of the object alongside their dynamic state vectors. It measures the "mass" needed to transform one density to another one, and the Wasserstein distance between two multivariate Gaussians can be expressed as [42]

$$d(\mathcal{N}_x, \mathcal{N}_{\hat{x}})^2 = ||m_x - m_{\hat{x}}||^2 + \mathrm{Tr}\left(P_x + P_{\hat{x}} - 2\sqrt{\sqrt{P_x} P_{\hat{x}} \sqrt{P_x}} \right), \tag{2.79}$$

where $\mathcal{N}_x = \mathcal{N}(m_x, sP_x)$ and m_x represents the center of the ellipse, and sP_x repsesents the extent. s is a scaling factor that relates to the tolerance region and is in this work (and in equation 2.79) used as $s = 1$. We used Tr to denote the trace operator.

3
Implementation

This chapter gives a walk-through of our implementation of the PMB filters, which was done in Python. It includes assumptions made for the data handling, the used code structure, different variations of the used filters and models, and the investigated scenarios. This chapter also includes values used for the design parameters and tuned parameters in the different parts of the code.

3.1 Assumptions

In the scope of this thesis, the focus was on highway situations. Therefore, it is reasonable to assume that that most objects have an almost constant velocity. Furthermore, the radar data is assumed to be 2D because the tracking is done for vehicles on rather flat highways and in this context the movement with relation to the z axis is negligible. This also enables us to better visualize the results. Since we are not considering slow-moving traffic or stop-and-go scenarios, we assume that all stationary points (relative to the ground, such as lightning poles, trees, etc) can be removed. The range rate of the data gathered by the radar is used to calculate the speed of the objects relative to the ground, where the object is considered stationary if its speed relative to the ground denoted by $v_{\text{obj}}^{\text{ground}}$, is lower than a certain threshold.

The radar is not located at the origin of the coordinate system of the vehicle, which means that we should find the coordinates of the measurements in the coordinate system of the ego vehicle, presented in Figure 3.2. The ego vehicle is the vehicle on which the radar is mounted. The field of view is a circular arc centered at the origin of the radar's coordinate system, see illustration in Figure 3.1. In our highway scenarios, the difference between the angle of the object relative to the radar (ψ) and the angle of the object relative to the car is negligible, as illustrated in Figure 3.2. This means that for speed calculations, we can assume that the object range rate relative to the radar denoted by r is equal to the object range rate relative to the ego vehicle. It is also assumed that the speed of the object perpendicular to the range rate direction is equal to zero because it can not be measured using the radar. This means that the $v_{\text{obj}}^{\text{ground}}$ can be calculated by adding the ego vehicle speed to the object range rate. Thus, the longitudinal speed component in the direction of the range rate v_1 and the latitudinal speed component in the direction of the range rate v_2 are calculated and then added to the object range rate to find the object's speed relative to the ground:

$$v_{obj}^{\text{ground}} = v_1 + v_2 + r = v_{ego,x}cos(\psi) + v_{ego,y}sin(\psi) + r. \tag{3.1}$$

Note that this equation is only valid within the field of view.

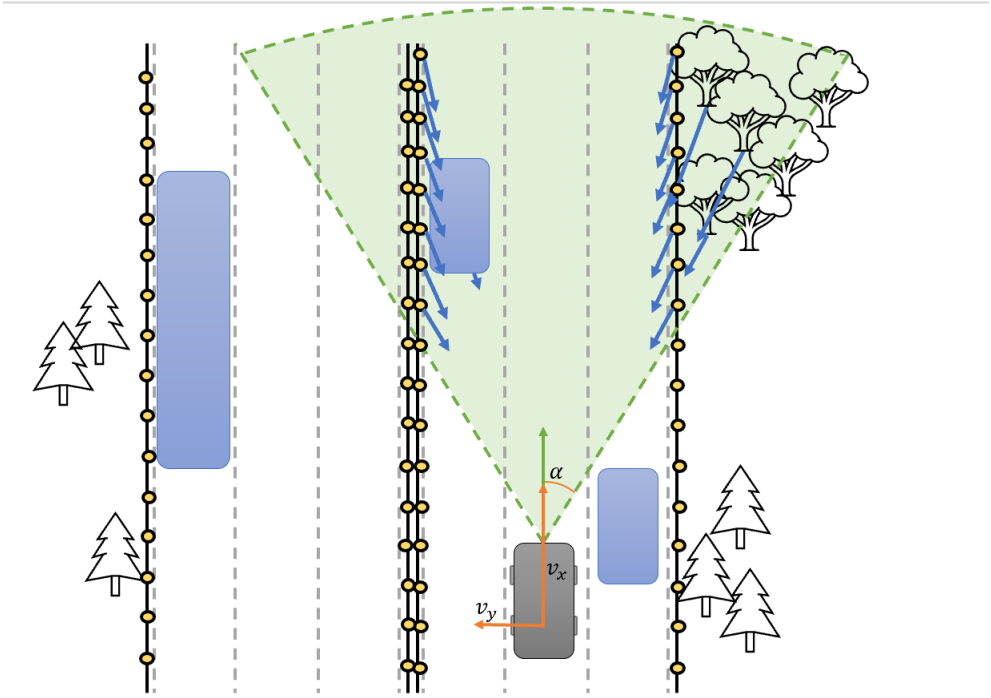


Figure 3.1: Figure showing a highway with an ego vehicle (grey), moving objects (blue), and stationary objects in form of trees and highway guard rails. The figure illustrates the field of view of the ego vehicle (green area surrounded by green dotted lines). The blue arrows are representing the range rate measured by the radar for each object. Thus the relative speed to the vehicle in the coordinate system of the radar. The objects without blue arrows are outside the field of view.

Another assumption made in this thesis is that the longest vehicle on the road is 25.25 meters, which is the truck length limitation in Sweden [43]. The clusters are thus adjusted to not exceed this length limit. Since each object in extended multiple object tracking can be represented by several measurements, each cluster can include one to infinity number of measurement points. Several clusters can also include the same measurement. But each cluster can only be assigned to one object, and one object can only be assigned to one cluster at each time step. Note that each assignment of a cluster to an object is a local hypothesis.

In a normal scenario, objects may enter and leave the field of view at all times. For the sake of simplicity, it can be assumed that the probability of detection and the probability of survival of an object is constant. We also assumed that the measurement noise can be chosen according to the sensor properties. More about the measurement noise assumption can be read in Section 3.5.

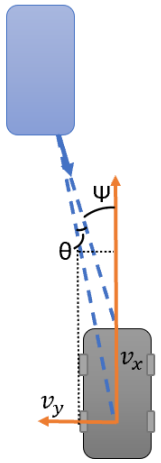


Figure 3.2: Figure showing a schematic of the vehicle speed and the object range rate as detected by the radar. The orange arrows represents the vehicle coordinate system.

3.2 States

Each estimated object is represented by a state consisting of a kinematic state vector, m_{est}, with position and velocity in two dimensions, together with a representation of the object's extent E_{est}.

$$m_{est} = \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix} \tag{3.2}$$

$$E_{est} = \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix} \tag{3.3}$$

As mentioned in Section 2.3.4, the objects are represented by elliptical shapes. An ellipse in a two-dimensional space can be represented by its semi-major axis l_1, its semi-minor axis l_2, and its angle relative to the x axis denoted by α_{rot}, see Figure 3.3. This information is extracted from the extent matrix by eigen decomposition as shown in (3.4),

$$E_{est} = R_{rot} \Sigma R_{rot}^T \tag{3.4}$$

Where R_{rot} is a rotation matrix and Σ is a diagonal matrix containing the eigenvalues of E_{est}:

$$R = \begin{bmatrix} \cos(\alpha_{rot}) & -\sin(\alpha_{rot}) \\ \sin(\alpha_{rot}) & \cos(\alpha_{rot}) \end{bmatrix}, \quad \Sigma = \begin{bmatrix} l_1^2 & 0 \\ 0 & l_2^2 \end{bmatrix}. \tag{3.5}$$

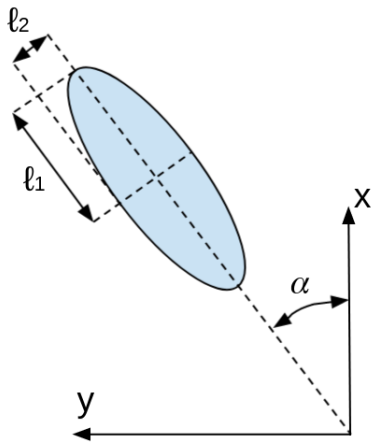


Figure 3.3: Figure depicting an ellipse in a 2D plane with semi-major axis l_1 and semi-minor axis l_2.

3.3 Used Filter Variations

Besides the assumptions, there are several different versions of the PMB filter and correlating models. In this sections, the used models and variations will be motivated and specified.

3.3.1 Filters

As described in Chapter 2, there are both PMBM filters and PMB filters, where the MBM is approximated as a single MB. The PMBM filter is a more complex version since all global hypotheses are saved over time steps and evaluated. This requires a lot of computational power and time, which is reduced by the approximation in the PMB filter. Since the aim of the filter is to be active in a vehicle, the runtime and complexity are very important factors. According to [1], the PMB filter does not decrease the accuracy significantly. The PMB filter is therefore the used filter in this thesis.

In Section 2.4.2, several ways to approximate MBMs to MBs were described. According to [1], the results are fairly similar between the different versions and the main difference tends to be when the signal-to-noise ratio is low and when objects are close to each other. For those cases, the Variational Multi-Bernoulli (VMB) yields better performance than the other implemented PMB and PMBM filters. Since the novelty of this thesis is the usage of real data, the result might differ and we have hence implemented three different versions of the filter, the Track Oriented PMB (TO-PMB), and the Variational Multi-Bernoulli consisting of the two versions; Most Likely Assignment (MLA-PMB) and Efficient Approximation of Feasible Set (EAFS-PMB).

Another filter variation is the usage of a data association algorithm. The implemen-

tation in this thesis uses an already implemented version of the Murty algorithm, namely fastmurty which finds the M assignments that yield the lowest cost (see used value for M in Table 3.2). More information about the fastmurty implementation can be found in [44].

3.3.2 Motion Model

Dependent on the type of data and the scenario used, different motion models might result in a very different performance of the algorithm. Since this thesis is focusing on highway data, the velocities of the tracked objects and the ego vehicle can be considered as constant which means that the Constant Velocity (CV) motion model is the most suitable model for our targets. Thus, a constant velocity motion model has been used.

3.3.3 Measurement Model

One measurement model was implemented and used throughout the evaluation of the filter, namely the PPP measurement model with the intensity function modeled as a random matrix model. Here, we recalculated the measurements into position vectors instead of using the measurements directly in the measurement vector. This allowed for the measurement model to be linear, which made it possible to eliminate measurements that were stationary using the range rate (as described in Section 3.1).

3.3.4 Birth Models

Two different birth models were used and compared in this thesis. One measurement-driven birth model, where all measurements are used as possible birth components, and one birth model with predefined fixed birth areas that are tuned to fit the data and in this case thus also realistic traffic scenarios. The concept of the predefined birth model is illustrated in Figure 3.4. As can be seen in the figure, more birth components are placed near the sensor since it is both more likely and more critical if an object appears in that region. The parameters θ and $n_{sections}$ can be tuned to fit the size of the field of view, where θ is the delta angle used to create uniformly angled lines in the field of view, see Figure 3.4, on which possible birth components are created with a certain interval. The interval is defined by $d = L_{FOV}/n_{sections}$, where L_{FOV} is the length of the field of view, and $n_{sections}$ is the number of birth components to add on each line. The lines are added from the center of the field of view (red dotted line in the figure) and out, with the sides of the field of view as the limit. The ellipses are representing the object extents and all measurements within a gating threshold representing the uncertainty of the position and the measurement covariance, are considered to belong to that object's birth component.

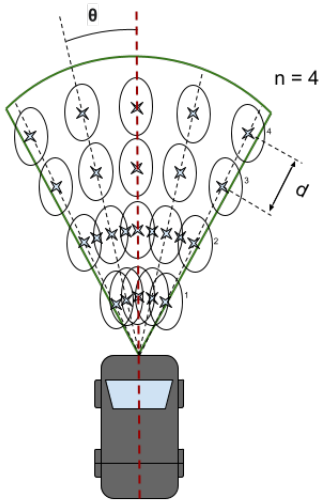


Figure 3.4: Figure illustrating the uniformly distributed birth model, with the tunable parameters θ and $n_{sections}$. In the example shown in the figure, $n_{sections} = 4$.

3.4 Algorithm Structure

The PMB filter contains many different parts, which are highly interconnected. To illustrate our implementation, we will go through our structure by presenting pseudo code. The main filter includes a loop over each time step with the general steps needed to perform filtering and evaluate the result, see Algorithm 4. In the algorithm, PPP includes both the state densities and the weights for the Poisson components. The same thing applies to the birth components. The variable *table* in the algorithm is the global hypothesis lookup table presented in Section 2.2.1. In the data pre-processing step, we need to load and prepare the dataset from its raw format into an organized and suitable structure for filtering purposes. The preparation of the data also includes changing the coordinate system of the detections from the radar coordinate system to the coordinate system of the ego vehicle and removing stationary points.

Algorithm 4 Implementation structure of the algorithms

```
1: Inputs: radar data, ground truth
2: Outputs: GOSPA value
3:
4: for every time step do
5:     measurements := Prepare data (radar data)
6:     if first time step then
7:         birth model := Create birth model
8:         PPP := Initiated as the birth model
9:         tracks := Initiated as empty matrices
10:    PPP, tracks := PMB update (measurements, PPP, tracks)
11:    state estimates := Calculate PMB estimates (tracks)
12:    PPP, tracks := Predict PMB (PPP, tracks, birth model)
13:    GOSPA value := GOSPA(state estimates, ground truth)
```

Algorithm 5 Implementation of the PMB update step

```
1: Inputs: measurements, PPP, tracks
2: Outputs: PPP, tracks
3:
4: Perform ellipsoidal gating
5: Perform clustering for gated measurements to get measurement clusters
6: Update the PPP with the in-gate clusters
7: Create local hypotheses by updating tracks and creating new ones
8: Create a global hypothesis look-up table
9:
10: if no clusters then
11:     Set all targets as misdetected
12:
13: for every partition do
14:     Construct a cost matrix
15:     Find M best assignments with (fast) Murty's algorithm
16:     Update the global hypothesis look-up table
17:
18: Prune global hypotheses with smaller weights than threshold
19: Prune local hypotheses with smaller existence probability than threshold
20: Perform reduction by merging similar global hypotheses
21: Update PPP with misdetections
22:
23: if there are several global hypothesis then
24:     Approximate the MBM as an MB
25:     Perform pruning and recycling
26: else
27:     Perform pruning and recycling
28:
29: Remove Poisson components with smaller weights than threshold and merge
    similar ones
```

The most computationally and theoretically heavy parts of the filter in Algorithm 4 are in the PMB update step. To give a clear view of what this step is doing, a pseudo code for this specific part is therefore shown in Algorithm 5. Note that this algorithm does not present input and outputs to the steps in the function. This is because the steps are highly interconnected and the inputs and outputs would rather create confusion than clarification.

In Algorithm 5, some variations are made for the different filter implementations. This is done in the step at the end of the algorithm where the MBM is approximated as an MB. To clarify how the filter implementations differ, see Algorithm 6.

Algorithm 6 Implementation of the MBM to MB approximation

```
1: Inputs: PPP, tracks, GHLT, MBM weights
2: Outputs: PPP, tracks, GHLT, MBM weights
3:
4: Perform track oriented merging (this is a base line needed for all variations)
5:
6: if TO then
7:     Do nothing since it is already done
8:
9: else if MLA then
10:    while Not converged do
11:        Use global hypotheses and tracks to compare approximated objects to
12:        updated objects
13:        Solve optimization problem for comparison between previous and new
14:        objects, to assign the "right local hypothesis to the right track" and
15:        find the best permutation of the local hypotheses to merge
16:        Comparing order between updated tracks and approximated tracks
17:        if Order not converged (for the objects in the global hypothesis), or cost
   not almost optimal then
18:            Perform merging to create new approximated objects
19:
20: else if EAFS then
21:    Extract weights for each local hypothesis
22:    while Not converged do
23:        Use global hypotheses and tracks to compare approximated objects to
24:        updated objects
25:        Solve transportation problem using cost matrix and extracted weights
26:        if Weights not similar (for local hypotheses), or cost not almost optimal
   then
27:            Perform merging to create new approximated objects
28:
29: Recycle Bernoullis with smaller existence probability than threshold
30: Delete these single target hypotheses
```

In Algorithm 6, note that an initial track oriented merging is a baseline for all three used filtering versions. Therefore, this is implemented for all filters.

3.5 Design Parameters

The filter contains a lot of different parameters that can be tuned to receive as good performance as possible for the used data. There are approximately 15 parameters to tune or choose a value for in the filter, which is why we have separated the parameters into two groups, tuning parameters and design parameters. The tuning parameters have been decided by performing line search, where different tuning parameters were grouped dependent on correlation. Each setting for the high priority

parameters was evaluated for all time steps for every sequence used for tuning. For those parameters within a tuning parameter group, the resulting evaluation compared to ground truth was saved in a cost matrix and the setting generating the lowest cost was adapted to the filter. The final tuning parameters for both the low, mid and high prioritized parameters, together with the tested interval, is presented in Table 3.1, and the design parameters and their chosen values are represented in Table 3.2. The upper and lower bounds of the tuning parameters, shown in Table 3.1, were set using knowledge about the filter, the data, and by trying varying numbers to narrow down the interesting area plus some marginal to be sure to include the optimal point. The number of values in each interval was chosen dependent on the parameter and was between 12 and 30 and typically log scaled.

Some parameters that can either be tuned or set are the additive measurement noise and the multiplicative measurement noise. For this thesis, the additive measurement noise has been used as the sensor properties. Other values were tested but the default sensor properties were considered as a good solution. Further, the multiplicative measurement noise was initially set as 1/4 according to [31], but was changed after tuning to a value that represents the spread of our measurements within the extent better, see Table 3.1.

Table 3.1: Tuning table including the tuned parameters and their corresponding tested interval, where LL is the lower limit and UL is the upper limit.

Tuning table					
Parameters	Limits		Chosen Value		
	LL	UL	MLA	TO	EAFS
Motion noise (q_{mot})	10^4	10^{10}	$280k$	$110k$	$170k$
Measurement noise, multiplicative (q_{meas_mn})	10^{-1}	10	0.4	0.4	0.4
Clutter rate (λ_c)	10^{-3}	10^3	2	2	2
Measurement rate prediction parameter (η)	0.1	20	1.1	1.1	1.1
Extent decay rate (τ)	1	200	100	100	100

Table 3.2: Table with design parameters, where the variables are set according to recommendations in papers and evaluation of filter decisions.

Design parameter table	
Parameter	Chosen Value
Gating size (d_g)	10 m
Upper cluster limit size (d_{max})	25.25 m
Lower cluster limit size (d_{min})	0.25 m
Cluster delta distance (δ_d)	1 m
Probability of detection (P_d)	0.9
Probability of survival (P_s)	0.99
Number of solutions from Murty, (M)	3
PPP weight pruning threshold (P_{PPP})	10^{-5}
Global hypothesis weight pruning threshold (P_w)	10^{-3}
Local hypothesis existence probability pruning threshold (P_r)	10^{-3}
Recycling threshold (τ_r)	10^{-2}

3.6 Evaluation Data

In order to compare different settings and examine the filter variations adequately, three different scenarios have been chosen for evaluation.

Scenario 1: The sequence is 25.25 seconds (1011 time steps) long. The ego vehicle is in the right lane with two vehicles in front of it in the same lane, one of which is 20 meters ahead of the ego vehicle and slightly to the right and one is 70 meters ahead, also slightly to the right. There is no oncoming traffic. From the point of view of the radar, one of the two vehicles is sometimes completely hidden behind the other one. None of the vehicles exit the FOV during this sequence.

Scenario 2: The sequence is 20 seconds (800 time steps) long. In the beginning of the sequence, there is only one vehicle in front of the ego vehicle. After about 2 seconds, a car enters the FOV from the right of the ego vehicle. It passes the ego vehicle and drives side by side with the vehicle that is in front, and then continues on. It leaves the front part of the FOV after 15 seconds. This means that this scenario depicts object birth and object death.

Scenario 3: The sequence is 17.5 seconds (700 time steps) long. There is a large truck with a trailer in front of the ego vehicle at a distance of about 50 meters. There is a lot of oncoming traffic on the left lane which enter and exit the FOV but they are not considered in this work.

3.7 Evaluation Methods

The evaluation of the result was done both visually and numerically. The visual evaluation was done for the different steps between clustering and object estima-

tion. Here, we mostly made sure that the different steps seemed reasonable with respect to the measurements and ground truth. The numerical evaluation was done by comparing the resulted object estimation to ground truth object positions and extents which are presented as ellipses, and the accuracy was evaluated using the GOSPA metric described in Section 2.5.1 with the Gaussian-Wasserstein distance in Section 2.5.2 for the different traffic situations described in Section 3.6. All different components in GOSPA, such as number of misdetections, false detections and the localization error was also evaluated individually to give a clearer view of the specifics in the filter performance. To evaluate the position without the extent, the RMSE was also calculated and compared in the x and y direction as in Figure 3.3. The evaluation considers tracking both single and multiple vehicles in highway traffic situations. Both vehicles staying in the field of view (but moving within it), objects staying in the same lane and objects entering and leaving the field of view.

Algorithm 7 Sandbox tracker

```
1: Inputs: json file (default sandbox_tracker.json)
2: Outputs: hdf5 file
3:
4: def ___init___:
5:     Define parameters you want to use
6:
7: def ___del___:
8:     Uses tt_ego_vehicle_wrapper to delete data
9:     (used in Step function)
10:
11: def run_ev_module:
12:     Converting ego vehicle data
13:     (used in Step function)
14:
15: def Step:
16:     Uses input data for the current time step
17:     (here's where we implemented our filter)
```

4

Results

In this chapter, the results from the different scenarios explained in Chapter 3 are presented. Since the chosen scenarios are quite different, the results will be presented and commented separately and then discussed. In the following sections, "MLA-U" denotes the MLA-PMB filter with the uniform birth model and "MLA-MD" denotes the MLA-PMB filter with a measurement-driven birth model. The same notation is used for the TO-PMB filter.

The evaluation is performed in Python on a laptop with an Intel(R) Core(TM) i7-7820HQ @ 2.90GHz $\times$ 8 CPU.

4.1 Scenario 1

The numerical evaluation for Scenario 1 is presented in Table 4.1 and 4.2, where the first table represents the results when using the uniform birth model, and the second one represents the results using the measurement driven birth model.

According to Table 4.1, thus comparing the filter performance using the uniform birth model, the EAFS-U gives the lowest numerical errors. Although, it is not the fastest, with the average run time of 0.8156 seconds. The TO-U, which takes on average 0.719 seconds to perform a filtering recursion, has the smallest average run time for each time step. This is reasonable considering the additional computational cost of the optimization and merging steps in the MLA-U and EAFS-U. The TO-U yields worse results than the two other filters based on GOSPA, but estimates the position of the objects better than MLA-U, considering the RMSE values for the longitudinal and latitudinal position.

Similarly to the performance using the uniform birth model, is the performance for the different filters for the measurement driven birth model. The general difference is that the run time is significantly decreased for all methods, where the TO still has the smallest average run time with 0.114 seconds.

Table 4.1: Table showing the mean GOSPA distance per time step (GO) with $c = 10$, mean Gaussian-Wasserstein per time step (GW), average number of false detections per time step (NF), average number of missed detections per time step (NM), RMSE of longitudinal position in meters (RPX), RMSE of latitudinal position in meters (RPY), RMSE of extent length in meters (REL), RMSE of extent width in meters (REW) and the average run time of each time step in seconds (T) when using the uniform birth model for Scenario 1.

	Scenario 1 (Uniform Birth Model)								
Filter Type	GO	GW	NF	NM	RPX	RPY	REL	REW	T
TO-PMB	6.818	3.252	**0.034**	0.678	**1.360**	**0.433**	1.030	0.794	**0.719**
MLA-PMB	6.463	**2.926**	**0.034**	0.674	1.389	0.440	0.903	**0.311**	0.720
EAFS-PMB	**6.190**	3.267	0.044	**0.541**	1.365	**0.433**	**0.884**	0.464	0.8156

Table 4.2: Table showing the mean GOSPA distance per time step (GO) with $c = 10$, mean Gaussian-Wasserstein per time step (GW), average number of false detections per time step (NF), average number of missed detections per time step (NM), RMSE of longitudinal position in meters (RPX), RMSE of latitudinal position in meters (RPY), RMSE of extent length in meters (REL), RMSE of extent width in meters (REW) and the average run time of each time step in seconds (T) when using the MD birth model for Scenario 1.

	Scenario 1 (MD Birth Model)								
Filter Type	GO	GW	NF	NM	RPX	RPY	REL	REW	T
TO-PMB	6.393	3.227	**0.065**	**0.549**	**1.365**	0.433	0.852	0.585	**0.084**
MLA-PMB	**6.382**	**3.197**	0.074	0.562	1.37	0.442	**0.83**	0.564	0.093
EAFS-PMB	6.392	3.268	**0.065**	0.565	**1.365**	**0.425**	0.865	**0.514**	0.094

As can be seen in Table 4.1 and Table 4.2, the risk of misdetection is larger than the risk of false detections. This can be explained by looking at the data, where it becomes clear that one vehicle sometimes gets occluded by the other vehicle from the radar point of view. This is illustrated in Figure 4.1, where it is clear that no cluster corresponds to the second vehicle. By the same reasoning the second vehicle sometimes gets partly occluded, causing measurement points in a smaller area, and also fewer measurement points since the object is further away. Therefore, the estimated extents get smaller.

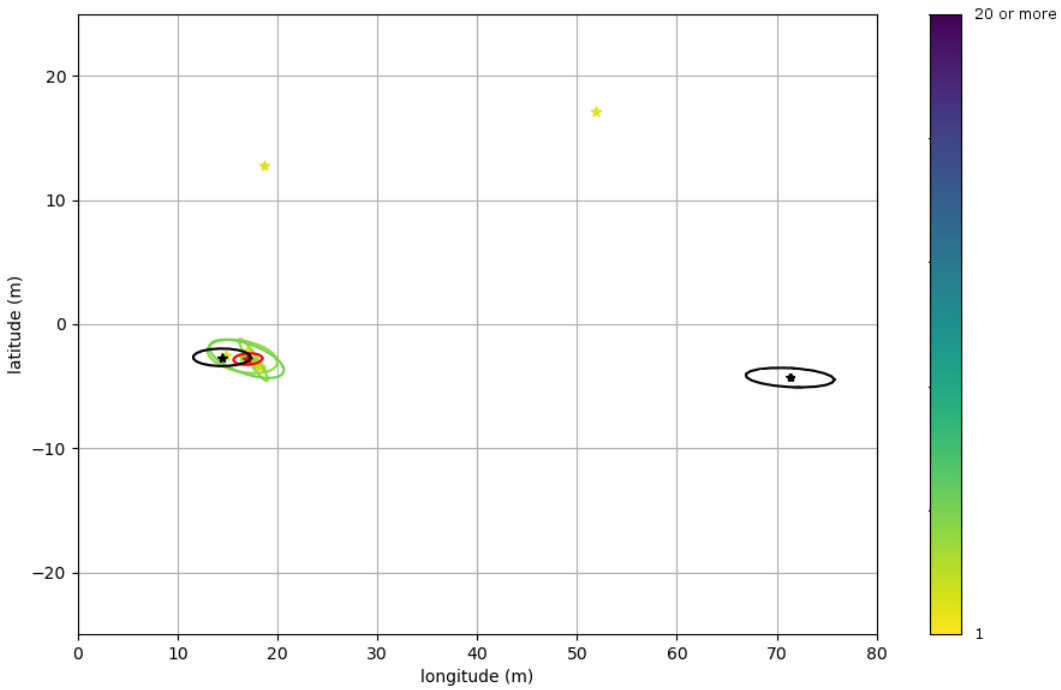


Figure 4.1: Figure showing the estimation plot for one time step. The black ellipses represent ground truth states, the red ellipse represents our estimate with MLA-PMB and the uniform birth model, and the remaining ellipses represent clusters of the measurements, where the color of the ellipse represents the number of measurement points in that cluster. Clusters containing more than 20 measurement points are represented as if they contained 20 measurement points. The stars are representing the center points in the ellipses.

4.2 Scenario 2

The numerical evaluation for Scenario 2 is presented in Tables 4.3 and 4.4, where the first table represents the results when using the uniform birth model, and the second one represents the results using the measurement driven birth model.

As can be seen in Table 4.3 and Table 4.4, TO-U tends to give the lowest numerical errors, while MLA-MD gives the lowest run time and is the second best considering the numerical errors. The relatively high number of false detections is due to two things: First is that the truck with a trailer in front of it is considered to be several objects by the filter, which causes the higher Gaussian-Wasserstein, and the second is that the algorithm estimates some clutter approximately 80 meters ahead to be objects for a few seconds.

Misdetections mostly happen in this scenario because of an object driving on the borderline of the FOV (to the right), where only the left front light is in the FOV

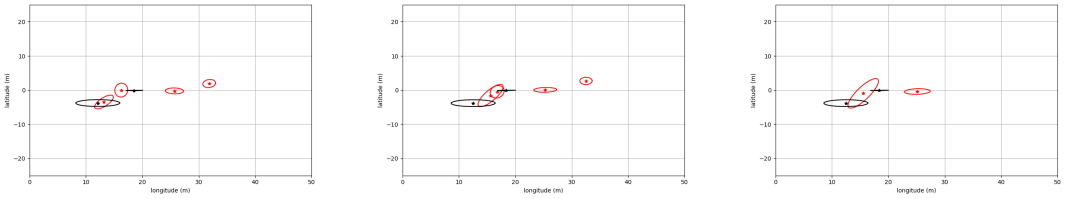
at time step 0. It is detected by the ground truth sensor but does not result in frequent enough measurements to be estimated by our filter. After approximately 2 seconds, one fourth of the vehicle is in the field of view and our filter estimates the vehicle occasionally. After six seconds, half of the vehicle is in the FOV and the estimation of the vehicle is stable. The vehicle passes and drives side by side with another vehicle. At a few seconds, they are driving very close, which results in the estimations to merge, which again causes misdetections. An illustration of this is shown in Figure 4.2.

Table 4.3: Table showing the mean GOSPA distance per time step (GO) with $c = 10$, mean Gaussian-Wasserstein per time step (GW), average number of false detections per time step (NF), average number of missed detections per time step (NM), RMSE of longitudinal position in meters (RPX), RMSE of latitudinal position in meters (RPY), RMSE of extent length in meters (REL), RMSE of extent width in meters (REW) and the average run time of each time step in seconds (T) when using the uniform birth model for Scenario 2.

	Scenario 2 (Uniform Birth Model)								
Filter Type	GO	GW	NF	NM	RPX	RPY	REL	REW	T
TO-PMB	**22.547**	6.609	**1.875**	**1.312**	**0.924**	**0.748**	**1.716**	1.243	7.333
MLA-PMB	24.691	6.503	2.055	1.583	0.988	0.818	3.025	**0.968**	**7.276**
EAFS-PMB	24.847	**6.367**	2.131	1.565	1.007	0.797	2.439	0.977	7.486

Table 4.4: Table showing the mean GOSPA distance per time step (GO) with $c = 10$, mean Gaussian-Wasserstein per time step (GW), average number of false detections per time step (NF), average number of missed detections per time step (NM), RMSE of longitudinal position in meters (RPX), RMSE of latitudinal position in meters (RPY), RMSE of extent length in meters (REL), RMSE of extent width in meters (REW) and the average run time of each time step in seconds (T) when using the MD birth model for Scenario 2.

	Scenario 2 (MD Birth Model)								
Filter Type	GO	GW	NF	NM	RPX	RPY	REL	REW	T
TO-PMB	26.028	7.659	2.161	1.512	0.883	0.760	2.449	1.992	1.944
MLA-PMB	**23.586**	**6.718**	**1.928**	**1.446**	0.812	**0.757**	**2.429**	**1.63**	**1.915**
EAFS-PMB	26.053	7.640	2.148	1.535	**0.810**	0.867	2.447	2.015	2.030



(a) Time step 1 **(b)** Time step 2 **(c)** Time step 3

Figure 4.2: Figures showing a sequence of time steps when two objects are estimated to be side by side, and gets estimated as one object. The black ellipses represent ground truth states, the red ellipses represent our estimates and the stars represents the center of the ellipses.

4.3 Scenario 3

The numerical evaluation for Scenario 3 is presented in Table 4.5 and 4.6, where the first table represents the results when using the uniform birth model, and the second one represents the results using the measurement driven birth model.

According to Tables 4.5, EAFS-U has the lowest GOSPA value, but it also has a rather large run time per time step (3.862 seconds) compared to EAFS-MD. EAFS-MD takes less time to run but results in a larger GOSPA and a higher number of missed and false detections on average. The MLA-MD seems to be the better alternative in the this case since it seems to achieve almost the same results as EAFS-U in less than a third of the time.

The TO-U has a significantly worse performance than the MLA-U and the EAFS-U, which is due to the higher number of missed and false detections on average. TO-MD has a better performance than TO-U in terms of the number of false and missed detections.

Table 4.5: Table showing the mean GOSPA distance per time step (GO) with $c = 10$, mean Gaussian-Wasserstein per time step (GW), average number of false detections per time step (NF), average number of missed detections per time step (NM), RMSE of longitudinal position in meters (RPX), RMSE of latitudinal position in meters (RPY), RMSE of extent length in meters (REL), RMSE of extent width in meters (REW) and the average run time of each time step in seconds (T) when using the uniform birth model for Scenario 3.

	Scenario 3 (Uniform Birth Model)								
Filter Type	GO	GW	NF	NM	RPX	RPY	REL	REW	T
TO-PMB	13.617	**6.567**	0.557	0.853	1.244	1.195	3.414	2.030	3.822
MLA-PMB	8.943	7.95	0.088	0.110	1.272	1.210	**2.736**	1.976	**3.779**
EAFS-PMB	**8.621**	7.793	**0.084**	**0.081**	**1.193**	**1.181**	3.200	**1.821**	3.862

Table 4.6: Table showing the mean GOSPA distance per time step (GO) with $c = 10$, mean Gaussian-Wasserstein per time step (GW), average number of false detections per time step (NF), average number of missed detections per time step (NM), RMSE of longitudinal position in meters (RPX), RMSE of latitudinal position in meters (RPY), RMSE of extent length in meters (REL), RMSE of extent width in meters (REW) and the average run time of each time step in seconds (T) when using the MD birth model for Scenario 3.

	Scenario 3 (MD Birth Model)								
Filter Type	GO	GW	NF	NM	RPX	RPY	REL	REW	T
TO-PMB	9.060	7.918	0.134	0.094	**1.247**	1.264	4.014	1.722	**1.031**
MLA-PMB	**8.654**	**7.797**	**0.083**	**0.088**	1.309	1.285	**3.936**	**1.333**	1.117
EAFS-PMB	9.082	7.818	0.147	0.106	1.258	**1.231**	4.769	1.658	1.114

The extent estimation for this scenario is not as good as in the first scenario, which is due to the size of the trailer and it's position relative to the ego vehicle. Most of the measurements in this case are generated from the side of the trailer, which is driving in front of the ego vehicle and to its right, which means that the filter considers it to be a long object with a narrow extent, as seen in Figure 4.3.

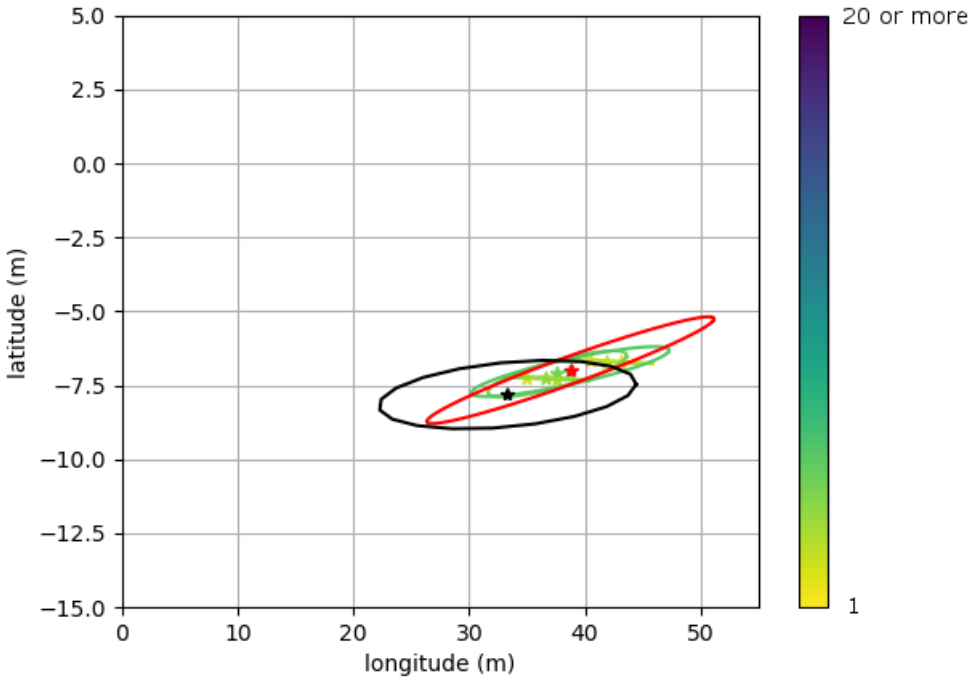


Figure 4.3: Figure showing the estimation plot at one time step for scenario 3. The black ellipse represents the ground truth state (in this case the trailer), the red ellipse represent our estimate, and the remaining ellipses represent clusters of the measurements, where the color of the ellipse represents the number of measurement points in that cluster. Clusters containing more than 20 measurement points are represented as if they contained 20 measurement points. The stars are representing the center points in the ellipses. It is evident that the measurements generated from the trailer do not represent the entire extent, so the resulting estimation is not similar to the true extent.

5
Discussion

To give a fair discussion of all relevant discussion points this thesis has created, we split this discussion chapter into four different sections where we first discuss the filter implementation. We continue by discussing the results and end this chapter with a discussion about future work.

5.1 Implementation

Several assumptions and simplifications were made in the implementation which can affect the results, for example, highway scenarios were chosen. Even if we tried to include different types of highway scenarios in the evaluation, not every highway scenario is evaluated. Hence the result can differ for other kinds of highway scenarios. Further, the provided ground truth data is measured by another, more accurate, sensor set which is mounted on the ego vehicle.

In this work, the range rate measurements were not used in the estimation since the single target measurement model was a simple two-dimensional model which only used the range and angle measurements which were then transformed to the longitudinal and latitudinal position. According to [45], radar measurements can be modeled using the range, angle, and range rate in three dimensions. This could improve both the position and the velocity estimates.

5.2 Discussion of Results

Of the three variations of the PMB filter that were investigated, the MLA-PMB seems to result in a good trade-off between performance and computational cost, which makes it a desirable alternative in automotive applications, where efficiency is as important as accuracy. The uniform birth model and the measurement-driven birth model yielded different results based on filter type. Between the two birth models, the smallest GOSPA was often achieved with the uniform birth model, but the uniform model was considerably slower. None of the implementations are fast enough to be used in real-time considering the sampling time, so if the filter is going to be implemented in a vehicle, some run time optimization needs to be done. If the filter using the uniform birth model can be made more efficient so that it runs at the same rate as the sampling rate or faster, then the uniform birth model is recommended. Otherwise, the measurement-driven birth model is a rather good alternative, which can have a run time that is about 3 to 8 times smaller than for

the uniform birth model, without sacrificing the estimation accuracy so much. For all scenarios, the combination of the MLA-PMB filter and the MD birth model resulted in a good performance when considering both the object estimation and the computational cost.

The MLA-PMB and the EAFS-PMB filters are likely to have a longer run time than the TO-PMB, which is reasonable considering that they first perform TO merging and then use the result as an initial guess for the optimization calculations. The optimization calculations are then used to find the optimal way to merge the Bernoullis. For the second and the third scenarios, the MLA-PMB filter has a smaller run time than TO-PMB in some cases. This can be because the TO-PMB filter is based on the assumption that all of the local hypotheses are independent and merges the local hypotheses for each track instead of considering all of the local hypotheses across all tracks and merging them by performing an optimization. This means that there is the possibility that some Bernoullis that would have been merged in MLA-PMB never get merged and this results in more Bernoulli components for TO-PMB. Thus, TO-PMB will have the additional computational cost for performing gating for these Bernoullis, updating them, and creating local hypotheses, and the run time will be larger. There is also a higher potential for the TO-PMB to detect one object as several overlapping objects compared to the other filters due to the local hypothesis independence assumption.

The accuracy of the objects' extent estimation varies depending on the position of the objects relative to the vehicle. However, the estimated objects' shapes are generally different from the true shapes. This is firstly due to the assumption that all objects are ellipse-shaped and the measurements are Gaussian-distributed over the area of the ellipse. This means that the true distribution of the measurements over the object is not considered, since the measurements would be more dense around the contours and less dense towards the center of the objects. The other reason is that the measurements are from only one radar and determining the shape of an object based on measurements from one or two sides of the object, dependent on the positioning relative to the radar, is not an easy task. From the results, it gets clear that the length of the objects is not estimated as accurately as the width of the objects. This is reasonable since the measurements collected by the front-looking radar are often the result of radar detections from the back of the vehicles which are driving in front of the ego vehicle and the full length is not completely observable. Thus, the estimation of the length is often more difficult. This is especially true if the object is too close to or too far from the vehicle. In the first case, the length is estimated to be almost similar to its true value and in the second case, the object is assumed to be smaller than it actually is. This is because the number of measurements decreases as the object gets further from the radar, and the extents are estimated based on the spread of the measurements. Furthermore, it is possible for some objects to be occluded if there are other objects directly in front of the radar.

5.3 Future Work

As discussed in Section 5.2, objects further away from the sensor tend to have fewer measurement points. This makes the extents of the objects likely to be smaller. A possible extension of this work could therefore be to extend the filter to remember the size of the objects as they move away from the ego vehicle so that they do not shrink in terms of size when moving away from the sensor.

In this work, it was assumed that the generated measurements were Gaussian-distributed on the surface area of the objects. This assumption does not hold in reality since the measurements are most likely to be detections from the side of the object that is facing the radar and not exactly on the contour. This means that there are no observations of the occluded side of the object. Thus, a more complicated measurement model can be chosen to represent both the occluded area and the specific distribution of the measurements which are often dense around the contour area and sparse in the middle of the object. One such model is presented in [46].

In Table 4.1 to Table 4.6, it is clear that neither of the filters detect all objects at all times. This can of course be very dangerous in a real-world scenario. One way to decrease the risk of misdetections is to implement a track manager to better keep track of detected objects over time. The idea is then that the track manager classifies the detected objects based on their position, velocity, and heading considering that objects are not likely to jump from one place to another or disappear abruptly from the FOV. The track manager can be especially helpful in cases of occlusion. But it can also be beneficial for false detections (which can be as dangerous if the ego vehicle emergency brakes for no reason) if implementing the track manager so that an object for example needs to be detected for a certain number of time steps before being added as an object in the track manager.

6
Conclusion

When developing autonomous vehicles, objects around the autonomous vehicle need to be tracked. Extended Multiple Object Tracking (EMOT) can be used to estimate the surrounding objects' position and shapes. In this work, filtering methods based on the Poisson Multi-Bernoulli Mixture conjugate prior were studied in order to choose an appropriate filter that takes radar data and solves the extended multiple target tracking problem efficiently for the purpose of tracking vehicles on the highway. The Poisson Multi-Bernoulli (PMB) filter was chosen due to its trade-off between high accuracy and low computational cost. Three variations of this filter, the Track-Oriented PMB (TO-PMB), the Most Likely Assignment PMB (MLA-PMB), and the Efficient Approximation of Feasible Set PMB (EAFS-PMB), were implemented and their performances were compared. Additionally, for each filter, two birth models were investigated: the uniform birth model, and the measurement-driven birth model. The provided measurements were collected by one radar mounted on a vehicle while driving on highways. The ground truth data for the position and extent of the objects was also collected by a separate reference system.

It was concluded that all three implemented filters performed well, where the MLA-PMB filter generally yields slightly better results. The filter manages to detect objects fairly well, although the extent estimations are not entirely accurate, due to one single radar's difficulty to correctly simultaneously observe the full length and width of the objects. The extent estimates can be improved by using multiple radars at different angles, as well as other sensors. For all filters, the measurement-driven birth model resulted in a much lower computational cost while maintaining good performance. It is deemed to be a good choice for a real-time system that needs to fulfill strict run-time budgets.

Other ways to further improve the filters were suggested such as representing the spatial distribution of the objects with a more advanced model, evaluate several different motion and measurement models, or implementing a track manager to classify and keep track of the objects.

Bibliography

[1] Y. Xia, K. Granström, L. Svensson, and M. Fatemi, "Extended Target Poisson Multi-Bernoulli Filter," *arXiv preprint arXiv:1801.01353*, 2019.

[2] National Highway Safety Administration, U.S. Department of Transportation, "Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey." [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812115

[3] S. Blackman, "Multiple Hypothesis Tracking for Multiple Target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, 2004.

[4] R. Mahler, "Multitarget Bayes Filtering via First-Order Multitarget Moments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.

[5] Á. F. Garcá-Fernández, Y. Xia, K. Granström, L. Svensson, and J. L. Williams, "Gaussian Implementation of the Multi-Bernoulli Mixture Filter," in *2019 22th International Conference on Information Fusion (FUSION)*, 2019, pp. 1–8.

[6] Á. F. Garcá-Fernández, J. L. Williams, K. Granström, and L. Svensson, "Poisson Multi-Bernoulli Mixture Filter: Direct Derivation and Implementation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 4, pp. 1883–1901, 2018.

[7] J. Williams, "An Efficient, Variational Approximation of the Best Fitting Multi-Bernoulli Filter," *IEEE Transactions on Signal Processing*, vol. 63, March 2014.

[8] J. Tollich, S. Funken, and F. Opitz, "Gaussian Poisson Multi Bernoulli Mixture Filter and its Applications," in *2020 21st International Radar Symposium (IRS)*, 2020, pp. 167–172.

[9] M. Beard, S. Reuter, K. Granström, B. Vo, B. Vo, and A. Scheel, "A Generalised Labelled Multi-Bernoulli Filter for Extended Multi-Target Tracking," in *2015 18th International Conference on Information Fusion (Fusion)*, 2015, pp. 991–998.

[10] S. Reuter, A. Danzer, M. Stübler, A. Scheel, and K. Granström, "A Fast Implementation of the Labeled Multi-Bernoulli Filter Using Gibbs Sampling," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 765–772.

[11] C. Lundquist, K. Granström, and U. Orguner, "An Extended Target CPHD Filter and a Gamma Gaussian Inverse Wishart Implementation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 472–483, 2013.

[12] S. Reuter, B. Vo, B. Vo, and K. Dietmayer, "The Labeled Multi-Bernoulli Filter," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3246–3260, 2014.

[13] Y. Xia, K. Granström, L. Svensson, and Á. F. Garcá-Fernández, "Performance Evaluation of Multi-Bernoulli Conjugate Priors for Multi-Target Filtering," in *2017 20th International Conference on Information Fusion (Fusion)*, 2017, pp. 1–8.

[14] K. Granström, M. Baum, and S. Reuter, "Extended Object Tracking: Introduction, Overview, and Applications," *ISIF Journal of Advances in Information Fusion*, vol. 12(2), Dec. 2017.

[15] S. Pang and H. Radha, "Multi-Object Tracking Using Poisson Multi-Bernoulli Mixture Filtering For Autonomous Vehicles," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 7963–7967.

[16] K. Koch, *Introduction to Bayesian Statistics*. Springer Berlin Heidelberg, 2007. [Online]. Available: https://books.google.se/books?id=-Jc93GraPIwC

[17] W. Bolstad and J. Curran, *Introduction to Bayesian Statistics*. Wiley, 2016. [Online]. Available: https://books.google.se/books?id=BxfkDAAAQBAJ

[18] S. Särkkä, *Bayesian Filtering and Smoothing*, ser. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2013.

[19] V. Feller and W. Feller, *An Introduction to Probability Theory and Its Applications, Volume 1*, ser. A Wiley publication in mathematical statistics. Wiley, 1968. [Online]. Available: https://books.google.se/books?id=ZfFQAAAAMAAJ

[20] A. Gut, *The Multivariate Normal Distribution*. New York, NY: Springer New York, 1995, pp. 119–148. [Online]. Available: https://doi.org/10.1007/978-1-4757-2431-8_6

[21] L. Hammarstrand. State Space Models. [Online]. Available: https://www.youtube.com/watch?v=4d3SLcFGoAc&list=PLTD_k0sZVYFqjFDkJV8GE2EwfxNK59fJY&index=13

[22] T. Matsuzaki, H. Kameda, S. Tsujimichi, and K. Kosuge, "Maneuvering Target Tracking Using Constant Velocity and Constant Angular Velocity Model," in *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics*, vol. 5, 2000, pp. 3230–3234 vol.5.

[23] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "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*, ser. KDD'96. AAAI Press, 1996, pp. 226–231.

[24] H. W. Kuhn, "The Hungarian Method for the Assignment Problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109

[25] E. Fortunato, W. Kreamer, Shozo Mori, Chee-Yee Chong, and G. Castanon, "Generalized Murty's Algorithm With Application to Multiple Hypothesis Tracking," in *2007 10th International Conference on Information Fusion*, 2007, pp. 1–8.

[26] X. He, R. Tharmarasa, M. Pelletier, and T. Kirubarajan, "Accurate Murty's Algorithm for Multitarget Top Hypothesis Extraction," in *14th International Conference on Information Fusion*, 2011, pp. 1–8.

[27] K. Granström, M. Fatemi, and L. Svensson, "Poisson Multi-Bernoulli Mixture Conjugate Prior for Multiple Extended Target Filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 1, pp. 208–225, Feb 2020. [Online]. Available: http://dx.doi.org/10.1109/TAES.2019.2920220

[28] T. Li, J. Lu, W. Liu, and J. M. Corchado, "Approximate Gaussian Conjugacy: Parametric Recursive Filtering Under Nonlinearity, Multimodality, Uncertainty, and Constraint, and Beyond." *Frontiers Inf Technol. Electronic Eng.*, vol. 18, pp. 1913–1939, 2017.

[29] Y. Xia, K. Granström, L. Svensson, and Á. F. Garcá-Fernández, "An Implementation of the Poisson Multi-Bernoulli Mixture Trajectory Filter via Dual Decomposition," in *2018 21st International Conference on Information Fusion (FUSION)*, 2018, pp. 1–8.

[30] F. E. Harris, "Chapter 9 - Gamma Function," in *Mathematics for Physical Science and Engineering*, F. E. Harris, Ed. Boston: Academic Press, 2014, pp. 325–347. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128010006000092

[31] M. Feldmann, D. Fraenken, and W. Koch, "Tracking of Extended Objects and Group Targets Using Random Matrices," *Signal Processing, IEEE Transactions on*, vol. 59, pp. 1409–1420, May 2011.

[32] L. Svenssson and K. Granström. Reduction of Local and Global Hypothesis. [Online]. Available: https://www.youtube.com/watch?v=s7xC3ciwWXc

[33] S. K. Singh, M. Premalatha, and G. Nair, "Ellipsoidal Gating For an Airborne Track While Scan Radar," in *Proceedings International Radar Conference*, 1995, pp. 334–339.

[34] R. P. Mahler, *Bayesian Filtering and Smoothing*, ser. Advances in Statistical Multisource-Multitarget Information Fusion. Artech House, 2014.

[35] T. Cover and J. Thomas, *Elements of Information Theory*. Wiley, 2012. [Online]. Available: https://books.google.se/books?id=VWq5GG6ycxMC

[36] L. Svensson, D. Svensson, M. Guerriero, and P. Willett, "Set JPDA Filter for Multitarget Tracking," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4677–4691, 2011.

[37] P. Pan, "A Fast Simplex Algorithm for Linear Programming," *Journal of Computational Mathematics*, vol. 28, no. 6, pp. 837–847, 2010. [Online]. Available: http://www.jstor.org/stable/43693619

[38] K. Granström and U. Orguner, "On the Reduction of Gaussian inverse Wishart Mixtures," in *2012 15th International Conference on Information Fusion*, July 2012, pp. 2162–2169.

[39] J. Soch and C. Allefeld, "Kullback-Leibler Divergence for the Normal-Gamma Distribution," *arXiv preprint arXiv:1611.01437*, 2016.

[40] A. S. Rahmathullah, Á. F. García-Fernández, and L. Svensson, "Generalized Optimal Sub-Pattern Assignment Metric," in *2017 20th International Conference on Information Fusion (Fusion)*, 2017, pp. 1–8.

[41] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A Consistent Metric for Performance Evaluation of Multi-Object Filters," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, 2008.

[42] S. Yang, M. Baum, and K. Granström, "Metrics For Performance Evaluation of Elliptic Extended Object Tracking Methods," in *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Sept. 2016, pp. 523–528.

[43] Transportstyrelsen. Modulsystem. [Online]. Available: https://www.transportstyrelsen.se/sv/vagtrafik/Yrkestrafik/Gods-och-buss/Matt-och-vikt/Modulsystem/

[44] M. Motro and J. Ghosh, "Scaling Data Association for Hypothesis-Oriented MHT," in *2019 22th International Conference on Information Fusion (FUSION)*, 2019, pp. 1–8.

[45] V. B. Frencl and J. B. R. do Val, "Tracking with Range Rate Measurements: Turn Rate Estimation and Particle Filtering," in *2012 IEEE Radar Conference*, 2012, pp. 0287–0292.

[46] Y. Xia, P. Wang, K. O. E. Berntorp, L. Svensson, K. Granström, H. Mansour, P. T. Boufounos, and P. Orlik, "Learning-based Extended Object Tracking Using Hierarchical Truncation Measurement Model with Automotive Radar," *IEEE Journal of Selected Topics in Signal Processing*, pp. 1–1, 2021.