UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

Radar-Centered Multi-Object Tracking With Moving Vehicle Platforms: An Initial Investigation

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

Ethan To
Norman, Oklahoma
2025

Radar-Centered Multi-Object Tracking With
Moving Vehicle Platforms: An Initial Investigation

A THESIS APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Yan Zhang, Chair

Dr. Wei Sun, Co-Chair

Dr. Justin Metcalf

Acknowledgments

I am especially thankful to my advisors, Dr. Zhang and Dr. Sun, for their continual guidance, encouragement, and expertise. Their mentorship has been instrumental in shaping both the technical depth and broader direction of this work. I am deeply appreciative of their patience, high standards, and dedication to fostering an environment of curiosity and rigor that has greatly influenced my academic and professional growth.

Finally, I would like to thank my family for their unwavering love, patience, and support. Their constant encouragement has provided the foundation that allowed me to pursue this research with determination and focus. This achievement would not have been possible without their belief in me.

Abstract

Radar sensors mounted on moving vehicle platforms present a promising solution for maritime surveillance. However, these systems face challenges due to measurement clutter and variations in the sensing environment. This thesis presents an initial investigation into radar-centered multi-object tracking using the Gaussian Mixture Probability Hypothesis Density (GM-PHD) filter within the Random Finite Set (RFS) framework. The objective is to evaluate the GM-PHD filter as a robust and computationally efficient alternative to classical multi-target trackers in maritime radar applications. The proposed system utilizes density-based clustering algorithms, such as HDBSCAN, to aggregate radar detections into structured measurement sets suitable for PHD filtering. The GM-PHD filter recursively estimates the multi-target intensity function, inherently accounting for target births, deaths, and missed detections without requiring explicit data association. This enables reliable tracking under heavy clutter and varying detection conditions. Experimental analysis was conducted using publicly available maritime radar datasets, including the DLR MANV and DARC collections. The results show that the GM-PHD tracker achieves comparable root-mean-square error (RMSE) performance to established algorithms such as the Joint Probabilistic Data Association (JPDA) filter, without the need for initial target assumptions or data association. These findings indicate that PHD-based radar tracking provides a strong foundation for future extensions toward radar-based Simultaneous Localization and Mapping (SLAM) on moving platforms.

Notation

Table: Notation Summary

Symbol	Description
$\mathbf{x}_k$	Target state vector at time step k
$\mathbf{z}_k$	Measurement vector or set at time k
$Z_{1:k}$	Collection of all measurements up to time k
X_k	Multi-object state represented as a Random Finite Set (RFS)
Z_k	Multi-object measurement set at time k
$p(\mathbf{x}_k \mid Z_{1:k})$	Posterior probability density of the state
$p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$	State transition probability (motion model)
$p(\mathbf{z}_k \mid \mathbf{x}_k)$	Likelihood function (measurement model)
p_S	Target survival probability
p_D	Probability of detection
$v_k(\mathbf{x})$	Probability Hypothesis Density (PHD) intensity function at time k
$\gamma_k(\mathbf{x})$	Birth intensity function of new targets
$\beta_k(\mathbf{x} \mid \zeta)$	Spawn intensity from existing target ζ
$f_{k\mid k-1}(\mathbf{x} \mid \zeta)$	Single-target transition (motion) density

Symbol	Description
$g_k(Z_k \mid X_k)$	Multi-object likelihood function
$w_k^{(i)}$	Weight of the i-th Gaussian component in the GM-PHD filter
$m_k^{(i)}$	Mean of the i-th Gaussian component
$P_k^{(i)}$	Covariance matrix of the i-th Gaussian component
J_k	Total number of Gaussian components at time k
T_P	Pruning threshold for Gaussian components
T_M	Merging threshold for Gaussian components
c_{cl}	Clutter spatial density
$\mathcal{N}(\mathbf{x}; \mathbf{m}, P)$	Gaussian distribution with mean $\mathbf{m}$ and covariance P
$\mathcal{W}(X; \nu, V)$	Wishart distribution with ν degrees of freedom and scale V
$\mathcal{IW}(X; \nu, V)$	Inverse Wishart distribution with parameters ν and V
$\mathbf{F}_k$	State transition matrix
$\mathbf{H}_k$	Measurement matrix or Jacobian (for nonlinear models)
$\mathbf{Q}_k$	Process noise covariance matrix
$\mathbf{R}_k$	Measurement noise covariance matrix
K_k	Kalman gain matrix
$\mathbf{P}_{k\|k-1}$	Predicted covariance of the state estimate
$\mathbf{P}_{k\|k}$	Updated covariance of the state estimate
$d^{(c)}(x, y)$	Capped distance used in OSPA metric
$\mathrm{OSPA}_p^c(X, Y)$	Optimal Subpattern Assignment metric of order p, cutoff c

Symbol	Description
$\rho(\mathbf{x})$	Clustering density or point intensity function
ϵ, μ	DBSCAN parameters: radius and minimum points per cluster
$\mathbf{R}$, θ	Range and bearing measurements (polar coordinates)
$\mathbf{u}_k$	Control input vector (for SLAM motion model)
$\mathbf{X}_{0:t}$	Sequence of robot poses up to time t
$\mathbf{m}$	Map or set of landmark states
$P(\mathbf{X}_{0:t}, \mathbf{m} \,\vert\, \mathbf{Z}_{1:t}, \mathbf{U}_{1:t})$	Joint SLAM posterior over trajectory and map
ϕ_k	Vehicle orientation (heading angle)
$\mathbf{x}_r$	Robot/vehicle pose (position and heading)
N_k	Estimated number of targets at time k (from $\int v_k(x)\,dx$)

Notes: All random variables are bold lowercase (e.g., $\mathbf{x}$) for vectors and bold uppercase (e.g., $\mathbf{X}$) for sets or matrices. Probability densities are defined over continuous Euclidean spaces. Units follow SI conventions: meters [m], radians [rad], and seconds [s].

Contents

List Of Tables

List Of Figures

Chapter 1

Radar Navigation in Complex Environments

1.1 Overview

Autonomous navigation in maritime environments requires robust perception under conditions like rain, fog, and sea clutter. Vision and LiDAR systems often degrade in this environment, while radar maintains reliable perception. However, radar perception returns multiple measurements from a single target, and the number of targets changes as targets enter and leave the scene. Therefore, radar perception is a multi-object tracking (MOT) problem.

Traditional tracking approaches often struggle in this context because of combinatorial bottlenecks due to explicit data association. To overcome these limitations, this thesis investigates the Gaussian Mixture Probability Hypothesis Density (GM-PHD) filter. Rooted in Random Finite Set (RFS) theory, this framework treats the collection of targets and measurements as single random sets. This allows for the joint estimation of target states and cardinality without explicit data association [3]. This thesis evaluates the GM-PHD filter as a robust and efficient algorithm for maritime environments.

1.2 Advantages of Radar

Radar sensors are used in autonomous navigation systems to measure the surrounding environment under various weather conditions. Other solutions utilizing LiDAR and cameras may offer higher visual fidelity under ideal conditions, but rain, darkness, fog, sea spray, and airborne particles such as dust significantly degrade their performance [4][5]. Radar also provides reliable range measurements that remain consistent in the presence of clutter and dynamic sea conditions. Modern camera systems can estimate motion or depth using stereo vision, but still rely on stable lighting and clear visibility. In the ocean, those conditions are not guaranteed.

Together, these characteristics motivate the use of radar as the primary sensor throughout this thesis, forming the physical basis for the MOT and SLAM formulations.

1.3 Extended Target Tracking

In practical radar systems, objects often return multiple detections. In many traditional tracking algorithms, there is a point target assumption. To overcome this limitation, an *Extended target model* is used. The extended target concept allows clusters of detections to be treated as a group belonging to a single target.

Successfully tracking extended targets is the first step in building a successful SLAM system. The following topics are problems that must be solved when implementing extended target tracking: data association and pose estimation.

1.3.1 Data Association

Data association is one of the most challenging problems in target tracking. In modern SLAM algorithms, Random Finite Sets (RFS) are used to address this issue. When a mobile robot follows different trajectories in an environment, the observed features it detects appear in a different order for each trajectory. Traditionally, these features would need to be reordered to align with the respective static features of the environment (Map Set). However, with the RFS approach, both the Map Set M and the feature set Z are represented as finite sets that assume no specific ordering of features. Modern SLAM approaches often leverage Bayesian statistics and RFS theory, allowing us to model the problem as a density function rather than as specific points [6]. As a result, data association between measurements and features is handled implicitly.

1.3.2 Clustering

Clustering serves as a preprocessing step to group these detections into a collection of individual targets. While methods such as k-means are often used, they require prior knowledge of the number of clusters and are sensitive to noise and outliers. Density based approaches, such as DBSCAN or HDBSCAN, overcome these limitations by identifying clusters according to local point density without needing to predefine the number of clusters. This clustering-based preprocessing has been widely explored in recent MOT research [7][8][9].

1.4 Introduction of Simultaneous Localization and Mapping (SLAM)

1.4.1 History of SLAM

The history of SLAM dates back to the 1986 IEEE Robotics and Automation Conference, where many of the leading researchers in the field met and shared ideas about applying probabilistic methods to robotics navigation [10][11]. These discussions spawned many of the most important foundational SLAM papers, most notably

Randall C. Smith and Peter Cheeseman's "On the Representation and Estimation of Spatial Uncertainty" [12]. This paper introduces a method of estimating the global spatial uncertainty of an object (or objects) by using relative measurements to create a network of position estimates. Smith and Cheeseman utilize approximate transformations (AT), which are estimated mean relation of one coordinate frame relative to another and a covariance matrix that expresses the uncertainty of the estimate [12].

By defining a series of movements as a network of ATs, Smith and Cheeseman then aim to simplify the system by combining series and parallel ATs. Although ATs as a concept are no longer used in modern state estimation theory, their formulation and other ideas from this article (and other papers like it) paved the way for a singular essential concept for the basis of SLAM: all detected objects or landmarks in a given scene are "correlated with each other because of the common error in estimated vehicle location"[10][13].

This realization forms the main root of modern SLAM theory, as it necessitates an algorithm to perform two functions: track both the pose of the vehicle and the distribution of landmarks, while also updating the current estimations with each new sample. This initially appeared to place an upper limit on an algorithm's capability, as the number of landmarks one could track would be thresholded by the available computing power. It wasn't until the 1995 International Symposium on Robotics Research that this inhibitor was removed upon finding that a system's global estimation error will converge [10]. This was the final pillar to be established in the prelude to the SLAM problem formulation, which will be explained in detail in the next section.

1.4.2 Initial Problem Statement

SLAM is the process by which a moving platform is localized within an environment while concurrently establishing a map of its surroundings. The object in motion is classically described as a "robot" in many of the foundational papers, but this "robot"

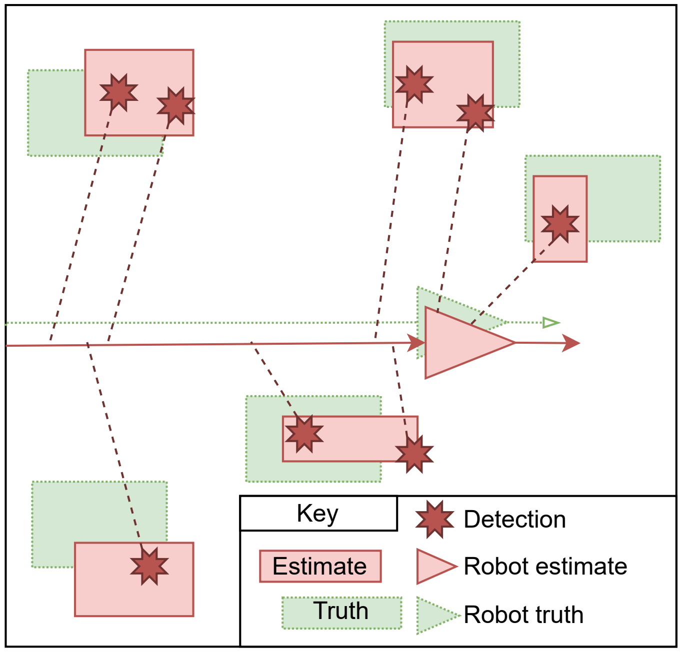

Figure 1.1: Illustration of the SLAM Problem

can take the form of any autonomous vehicle with movement and environment-sensing capabilities. Figure 1.1 shows the concept. [10].

As the robot moves throughout the environment, it will build a map of the environment by creating landmarks using the detections from the onboard sensors (in our case, this is radar). Additional detections allow for a heightened level of accuracy in both the robot estimate and the landmark estimates.

1.4.3 Pose Estimation

Estimating position and orientation (pose) is often solved through dead reckoning or particle filters. For radar-based SLAM, particle filters are effective because of their robustness in handling non-Gaussian noise and flexibility in uncertain environments. The robot's pose is estimated by generating a set of random possible states (particles), which are then refined over time using various matching techniques. For each set of measurements, the particle's predicted observation is compared to the actual radar measurements to assess the likelihood. Particles with high probability to the observed scan have increased weight, while those with low probability have decreased weight. These weights allow the particle filter to resample the particle distribution to the most probable. This recursive process of gathering radar measurements and refining the particles bring the estimated close to the robot's true position.

1.5 Organization of Thesis

Chapter 1 introduces the motivation and objectives of the research, emphasizing radar as a primary sensing modality for navigation and tracking in complex environments. It also outlines the concepts of Simultaneous Localization and Mapping (SLAM) and extended target tracking as the foundation of this work. Chapter 2 reviews radar fundamentals, including sensor models, measurement formulations, and clustering techniques such as K-means, DBSCAN, and HDBSCAN, along with performance metrics like the Optimal Subpattern Assignment (OSPA) measure. Chapter 3 provides the theoretical background for estimation and Bayesian filtering, covering key probabilistic concepts, recursive Bayesian estimation, and single-target filtering through the Kalman and Extended Kalman Filters. Chapter 4 introduces the Random Finite Set (RFS) framework for multi-object tracking and explains how the Probability Hypothesis Density (PHD) approximation simplifies the intractable multi-object Bayes filter. Chapter 5 develops the Gaussian Mixture PHD (GM-PHD) filter, describing its prediction, update, and component management processes for MOT. Chapter 6 extends the framework to model extended targets, introducing the Wishart and Inverse Wishart distributions and their applications to target extent estimation. Chapter 7 outlines the implementation of the algorithms in Python and discusses the marine datasets and preprocessing steps. Chapter 8 presents experimental results under single-target and multi-object scenarios using RMSE and OSPA metrics. Chapter 9 introduces the PHD-SLAM framework, and demonstrates its performance using synthetic radar data. Finally, Chapter 10 concludes the thesis and discusses potential directions for future work in radar-based SLAM and multi-object tracking.

Chapter 2

Fundamentals of Radar Tracking

2.1 Radar Sensor Model and Principles

Radar technology has evolved significantly since its first major applications during World War II, when it served as a critical tool for detection and navigation. Early radar systems relied on vacuum-tube transmitters and mechanically steered antennas, resulting in bulky and maintenance-intensive architectures with limited precision. Advances in microwave engineering and semiconductor technology have since enabled the development of compact, energy-efficient, and highly reliable solid-state radar systems that dominate modern commercial and maritime applications. Solid-state transmitters provide enhanced operational stability, extended service lifetimes, and improved waveform flexibility compared to traditional vacuum-tube designs, marking a major milestone in radar modernization [14] [15].

The objective of this thesis is to investigate radar as the primary sensing modality for perception and navigation of ocean-based unmanned platforms. Modern radar systems commonly employ pulsed or frequency-modulated continuous-wave (FMCW) waveforms to estimate the range and radial velocity of targets with high accuracy. For a point target, the relationship between target range R and the measured round-trip time delay T of the reflected pulse is given by

$$R = \frac{cT}{2} \tag{2.1}$$

where c denotes the speed of light. This relationship forms the basis of range estimation in pulsed radar systems and underpins most radar-based sensing techniques across terrestrial, airborne, and maritime domains. A simplified pulsed radar model can be described by the transmitted and received signals. The transmitted pulse is expressed as

$$s_t(t) = A_t \operatorname{rect}\left(\frac{t}{\tau}\right) \cos(2\pi f_c t)$$

where A_t is the transmission amplitude, τ is the pulse width, and f_c is the carrier frequency. After reflection from a target at range R, the received signal is modeled as

$$s_r(t) = A_r \operatorname{rect}\left(\frac{t - t_d}{\tau}\right) \cos[2\pi f_c (t - t_d)]$$

where A_r is the received amplitude and $t_d = \frac{2R}{c}$ represents the round-trip delay time. The received power is governed by the fundamental radar equation:

$$P_r = \frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 R^4} \tag{2.2}$$

where P_t is the transmitted power, G is the antenna gain, λ is the wavelength, and σ denotes the radar cross-section (RCS) of the target. Finally, the range resolution, the ability to distinguish two targets separated in range, is determined by

$$\Delta R = \frac{c\tau}{2} \tag{2.3}$$

where shorter pulse widths yield finer range discrimination. Together, these equations describe the fundamental behavior of pulsed radar systems and establish the physical basis for the sensing models utilized in this research [16].

2.1.1 Pulsed vs CW Radar

In pulsed radar, the transmitter emits high energy pulses separated by a pulse repetition interval. Between these pulses, the radar receives reflected signals from targets before the next pulse is transmitted. Using equation (2.3), the time delay between transmission and reception allows for the calculation of the target's range. Velocity is estimated by measuring the Doppler shift between the reflected and transmitted signals. However, the velocity measurements can become ambiguous due to aliasing if the Doppler frequency exceeds the system's Nyquist limit.

In contrast, Continuous Wave (CW) radar systems transmit and receive signals simultaneously. Due to the transmission being continuous, standard methods cannot be used to estimate range. Instead, range measurement requires the use of Frequency Modulated Continuous Wave (FMCW) techniques. One common method of determining the range is to linearly modulate the transmit signal. The frequency shift between the transmitted and received signal is the beat frequency. This beat frequency is proportional to the time delay so the target's range can be computed with the slope.

2.2 Measurement Model

In target tracking, the measurement model defines the probabilistic relationship between the system state and the sensor observations. While the state vector x_k typically contains kinematic variables such as position and velocity, the radar sensor only observes the position.

2.2.1 Linear Cartesian Model

When radar detections are pre-processed and transformed into a Cartesian frame (e.g., East-North-Up), the relationship between the state and measurement becomes linear:

$$z_k = H_k x_k + v_k \tag{2.4}$$

Assuming a standard state vector comprising position and velocity components $x_k = [x \quad y \quad \dot{x} \quad \dot{y}]^\top$, the measurement matrix H_k projects the state onto the position subspace:

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

While this linear structure allows for the use of standard linear estimators, it necessitates that the measurement noise covariance R_k be defined in the Cartesian frame. This transformation is often an approximation, as it assumes the sensor's uncertainty is uniform along the X and Y axes, masking the true range-dependent anisotropy of the radar sensor.

2.2.2 Nonlinear Polar Model

Raw radar measurements are natively acquired in polar coordinates (range r and azimuth θ). To preserve the true anisotropic nature of the sensor noise, the nonlinear observation function is used:

$$z_k = \begin{bmatrix} r \\ \theta \end{bmatrix} = h(x_k) + v_k = \begin{bmatrix} \sqrt{x^2 + y^2} \\ \arctan(y/x) \end{bmatrix} + v_k \tag{2.5}$$

Since $h(x_k)$ is nonlinear, recursive Bayesian filters require a linearization of this function to propagate the state covariance. This is achieved via the Jacobian matrix, discussed in Chapter 3. This nonlinear formulation allows the tracker to model range and bearing uncertainties independently, capturing the high range precision relative to angular resolution typical of marine radar.

2.3 Clustering Techniques

In many data processing and estimation problems, one of the central challenges is the data association problem, particularly when observations are closely spaced, overlapping, or partially observed. Clustering algorithms serve as a valuable pre-processing step in such contexts by grouping measurements spatially prior to the filter.

2.3.1 K-means Clustering

K-means is one of the most commonly used partition-based clustering algorithms due to its simplicity and computational efficiency. The algorithm was originally developed by Lloyd in 1957 for vector quantization [17], and the term "k-means" was later introduced by MacQueen in 1967 [18]. It aims to partition a dataset into k clusters by minimizing the total intra-cluster variance, typically measured as the sum of squared Euclidean distances between data points and their corresponding cluster centroids. The algorithm proceeds iteratively through the following steps:

1. Initialize k cluster centroids, often chosen randomly from the dataset.
2. Assign each data point to the nearest centroid based on a chosen distance metric (typically Euclidean).
3. Recalculate the centroids as the mean of all points assigned to each cluster.
4. Repeat steps 2 and 3 until convergence, typically defined as when cluster assignments no longer change.

Despite its wide usage, K-means has several notable limitations. It requires the number of clusters k to be specified in advance, which is often not known a priori.

Moreover, it assumes clusters are convex and isotropic (i.e., of similar size and shape), which may not hold true for complex or irregular datasets. The algorithm is also sensitive to outliers and the initial placement of centroids, potentially leading to suboptimal solutions.

2.3.2 Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

Traditional clustering approaches, such as K-means, require prior knowledge of the expected number of clusters and are often vulnerable to noise. [19] introduced DBSCAN, which addresses both the estimation of the number of clusters and the handling of noise by defining clusters based on point density. It operates using two parameters: ε, which specifies the maximum distance between points in a neighborhood, and μ, which defines the minimum number of points required to form a dense region. The DBSCAN algorithm proceeds as follows:

1. A random unvisited point is selected from the dataset.
2. The set of neighbors within distance ε is determined.
3. If the point has at least μ neighbors, it is classified as a core point and a new cluster is formed.
4. All reachable points from the core point are added to the cluster.
5. The process is repeated for all newly added points until the entire dense region is covered.
6. Points that do not have enough neighbors and are not reachable from any core point are classified as noise.

Although DBSCAN is effective in identifying clusters without requiring prior knowledge of the amount or shape, it can struggle when the data has varying densities. In such cases, a single choice of ε may be too small for sparse regions or too large for dense ones, leading to either fragmented clusters or the merging of distinct groups. These limitations motivate the use of hierarchical extensions such as HDBSCAN.

2.3.3 Hierarchical DBSCAN

HDBSCAN, introduced in [20], extends DBSCAN by constructing a hierarchical representation of clusters and extracting the most stable clusters from this hierarchy. It eliminates the need for a fixed ε parameter by exploring clustering structure across all possible distance scales. HDBSCAN uses the concept of mutual reachability distance to define the distance between points, incorporating both the density around each point and their pairwise distance. The algorithm proceeds as follows:

1. Compute the core distance for each point, defined as the distance to its *minimum samples*-th nearest neighbor.

2. Define the mutual reachability distance between pairs of points as the maximum of their individual core distances and their actual distance.

3. Construct a minimum spanning tree (MST) from the mutual reachability graph.

4. Generate a hierarchy of clusters by progressively removing the longest edges from the MST, forming a cluster tree.

5. Condense the cluster tree by evaluating cluster stability and extract the most persistent clusters as final results.

Unlike DBSCAN, HDBSCAN can identify clusters of varying densities and automatically determine the number of clusters without specified thresholds.

2.4 Performance Metrics

2.4.1 Optimal Subpattern Assignment (OSPA) Metric

In multi-object tracking (MOT), it is common to evaluate tracker performance by comparing the set of estimated object positions to the set of ground truth positions at each time step. Due to the inherently unordered and time-varying nature of object sets, conventional distance metrics (e.g., Euclidean or mean-squared error) are insufficient. The Optimal Subpattern Assignment (OSPA) metric, introduced by Schuhmacher, Vo, and Vo [21], provides a principled way to quantify both localization and cardinality errors between finite sets. Let $X = \{x_1, x_2, \ldots, x_m\}$ denote the ground truth set and $Y = \{y_1, y_2, \ldots, y_n\}$ the estimated set of object positions in $\mathbb{R}^d$. For $p \geq 1$ (order parameter) and $c > 0$ (cutoff parameter), the OSPA metric of order p and cutoff c is defined as:

$$\mathrm{OSPA}_p^c(X, Y) = \left(\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^{m} d^{(c)}(x_i, y_{\pi(i)})^p + c^p(n - m) \right) \right)^{1/p}, \tag{2.6}$$

where $m = |X|$, $n = |Y|$, and Π_n is the set of all permutations of $\{1, 2, \ldots, n\}$. The capped distance function $d^{(c)}$ is defined as:

$$d^{(c)}(x, y) = \min\left(c, \|x - y\|\right), \tag{2.7}$$

with $\|\cdot\|$ denoting the Euclidean norm. If $m > n$, the roles of X and Y are swapped to ensure symmetry. The metric penalizes both localization errors (via the distances between matched elements) and cardinality errors (via the unmatched elements, each contributing a fixed penalty of c^p). The OSPA metric satisfies the properties of a true metric on finite sets, including symmetry and the triangle inequality. It is especially useful in the context of Random Finite Set (RFS)–based filters because it measures the discrepancy between unordered sets of targets and estimates without labeling or ordering of elements.

Chapter 3

Single Target Tracking and Bayesian Foundations

3.1 Conditional Probability

Conditional probability quantifies the likelihood of an event occurring given that some other event has occurred. It provides the mathematical framework for reasoning about dependent events and forms the foundation of Bayesian inference and state estimation methods used throughout this thesis. Let A and B be two events with $\Pr(B) > 0$. The conditional probability of A given B is defined as

$$\Pr(A \mid B) = \frac{\Pr(A \cap B)}{\Pr(B)}.$$

This definition expresses the probability of A occurring given that B has occurred. Intuitively, conditioning on B restricts the sample space to outcomes in which B is true, and the resulting probability of A is then evaluated relative to this reduced space. For continuous random variables, the concept of conditional probability is expressed in terms of probability density functions. Let X and Y be continuous random variables with joint density $p_{X,Y}(x, y)$ and marginal densities $p_X(x)$ and $p_Y(y)$. The conditional density of X given $Y = y$ is defined as

$$p_{X|Y}(x \mid y) = \frac{p_{X,Y}(x, y)}{p_Y(y)}, \quad \text{for } p_Y(y) > 0.$$

This density describes the relative likelihood of X taking a specific value when Y is known to equal y. The relationship between the joint, marginal, and conditional densities can be expressed as

$$p_{X,Y}(x, y) = p_{X|Y}(x \mid y)\, p_Y(y) = p_{Y|X}(y \mid x)\, p_X(x),$$

which is central to Bayesian reasoning. Conditional probability allows prior knowledge or observations to be incorporated into probabilistic models, leading directly to the recursive estimation frameworks discussed in the following sections.

3.2 Probability Density Function

A probability density function (PDF) is a non-negative function $p(x)$ defined over a continuous sample space such as $\mathbb{R}^n$, such that the probability that a random variable X lies within a measurable region $A \subseteq \mathbb{R}^n$ is given by:

$$\Pr[X \in A] = \int_A p(x)\, dx$$

The PDF must satisfy the normalization condition:

$$\int_{\mathbb{R}^n} p(x)\, dx = 1$$

Intuitively, $p(x)$ describes the density of probability mass near point x, but not the probability of $X = x$ itself, since:

$$\Pr[X = x] = 0 \quad \text{for continuous } X$$

Instead, the probability of X lying in a small region is approximated. The smaller the region, the better the approximation.

3.3 Binomial Distribution

The binomial distribution models the number of successes in a fixed number of independent trials, where each trial has the same probability of success. Let n be the number of trials and p be the probability of success on each trial. Then, the probability of obtaining exactly k successes is given by the binomial probability mass function:

$$P(N = k) = \binom{n}{k} p^k (1 - p)^{n-k}, \tag{3.1}$$

where $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ is the binomial coefficient, representing the number of ways to choose k successful trials out of n. The mean of the binomial distribution is np, and its variance is $np(1 - p)$.

In the context of radar, this distribution characterizes the detection probability P_D of detecting a target in a single scan (replacing p). In that case, the probability P_D (replacing p) is the probability of accumulating k successful detections over a window of n consecutive scans. This means that each scan can be viewed as an independent trial for detecting the target, and the distribution describes how likely it is to observe the target exactly k times out of n opportunities.

3.4 Poisson Distribution

The Poisson distribution models the number of events occurring in a fixed interval of time or space when the events happen independently at a constant average rate. Let λ be the average number of events per interval. Then, the probability of observing exactly k events is given by:

$$P(N = k) = \frac{\lambda^k e^{-\lambda}}{k!}, \tag{3.2}$$

where k is a nonnegative integer and $\lambda > 0$ denotes the mean rate of occurrence. In multi-object tracking, the Poisson distribution serves as the standard probabilistic model for *clutter* (false alarms) and *target birth*. Specifically, the number of false alarms generated by sea clutter in a measurement volume is modeled as a Poisson-distributed random variable with a spatial density λ_C.

This choice of model is physically justified by the limiting behavior of the binomial distribution. As the number of trials $n \to \infty$ and the success probability $p \to 0$ while the product $\lambda = np$ remains constant, the binomial distribution converges to the Poisson distribution. In the context of radar, the surveillance region consists of a vast number of resolution cells (n), but the probability of a false alarm occurring in any specific cell due to noise (p) is extremely low. This mathematical property makes the Poisson distribution the natural choice for describing the cardinality of clutter returns and spontaneous target appearances, forming the stochastic foundation for the random finite set models discussed in subsequent chapters.

3.5 Recursive Bayesian Estimation

Recursive Bayesian Estimation provides the theoretical foundation for probabilistic state estimation in dynamic systems. It is used in filtering algorithms such as the Kalman filter, particle filters, and the Probability Hypothesis Density (PHD) filter. The objective is to estimate the state $\mathbf{x}_k$ at time step k, given a sequence of measurements $\mathbf{z}_{1:k} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k\}$.

The solution involves recursively computing the posterior distribution $p(\mathbf{x}_k|\mathbf{z}_{1:k})$, which represents the belief about the system state after incorporating all observations up to time k. This is accomplished through a prediction and update step.

3.6 Single Target Statistics

In classical Bayesian filtering, the state dynamics and measurement process of a single target observed by a single sensor are typically modeled as:

$$\mathbf{z}_k = h_k(\mathbf{x}_k) + \mathbf{w}_k, \tag{3.3}$$

$$\mathbf{x}_{k+1} = g_k(\mathbf{x}_k) + \mathbf{v}_{k+1}, \tag{3.4}$$

where $\mathbf{x}_k$ denotes the target state at time k, $\mathbf{z}_k$ is the observation, $h_k(\cdot)$ is the measurement function, and $g_k(\cdot)$ is the process model describing target dynamics. The vectors $\mathbf{w}_k$ and $\mathbf{v}_{k+1}$ represent zero-mean random noise associated with the measurement and motion models, respectively. To enable Bayesian estimation, these models are expressed in probabilistic form. The measurement model yields the likelihood function:

$$f_{\mathbf{Z},k}(\mathbf{z} \mid \mathbf{x}) = f_{\mathbf{w}_k}(\mathbf{z} - h_k(\mathbf{x})), \tag{3.5}$$

which describes the probability density of observing $\mathbf{z}$ given the state $\mathbf{x}$, with $f_{\mathbf{w}_k}$ denoting the density of the observation noise. Similarly, the motion model gives rise to the Markov transition density:

$$f_{k+1}(\mathbf{x}_{k+1} \mid \mathbf{x}_k) = f_{\mathbf{v}_{k+1}}(\mathbf{x}_{k+1} - g_k(\mathbf{x}_k)), \tag{3.6}$$

where $f_{\mathbf{v}_{k+1}}$ is the probability density function of the process noise $\mathbf{v}_{k+1}$. These probability models form the foundation for recursive Bayesian filtering.

3.6.1 Prediction Step

Given the posterior at the previous time step $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ and the motion model $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, the predicted distribution at time k is obtained via the Chapman-Kolmogorov equation:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})\, p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})\, d\mathbf{x}_{k-1}. \tag{3.7}$$

3.6.2 Update Step

When a new observation $\mathbf{z}_k$ becomes available, the prior is updated to the posterior using Bayes' rule:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)\, p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})}, \tag{3.8}$$

where $p(\mathbf{z}_k|\mathbf{x}_k)$ is the likelihood and the denominator is a normalizing constant:

$$p(\mathbf{z}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{z}_k|\mathbf{x}_k)\, p(\mathbf{x}_k|\mathbf{z}_{1:k-1})\, d\mathbf{x}_k. \tag{3.9}$$

This recursive process continues with each new measurement. Recursive Bayesian estimation avoids storing past data, making it computationally tractable for real-time applications such as tracking and SLAM. However, exact computation is intractable except for linear Gaussian models (as in the Kalman filter), motivating approximations such as Gaussian mixtures.

3.7 Kalman Filtering

The foundation of modern SLAM algorithms is rooted in the principles of the Kalman Filter. This section explores its core equations, which form the basis for numerous state estimation and mapping techniques.

3.7.1 State Prediction

The general discrete-time motion model is [22],

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_{k-1}\,\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{v}_{k-1}, \tag{3.10}$$

where $\hat{\mathbf{x}}_{k|k-1}$ is the predicted state estimate, $\hat{\mathbf{x}}_{k-1|k-1}$ is the posterior estimate from the previous time step, and $\mathbf{F}_{k-1}$ is the state transition matrix that represents the motion model. The process noise $\mathbf{v}_{k-1}$ is modeled as zero-mean, independent, additive Gaussian noise. The state transition matrix propagates the previous estimate forward according to the assumed state dynamics, and the additive noise models uncertainty and unmodeled effects in the motion.

3.7.2 Covariance Prediction

The state covariance is propagated forward as [22],

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1}\,\mathbf{P}_{k-1|k-1}\,\mathbf{F}_{k-1}^{\top} + \mathbf{Q}_{k-1}, \tag{3.11}$$

where $\mathbf{P}_{k-1|k-1}$ is the posterior covariance from the previous time step, $\mathbf{F}_{k-1}$ is the same state transition matrix used in the motion model, and $\mathbf{Q}_{k-1}$ is the process noise covariance that corresponds to $\mathbf{v}_{k-1}$.

The term $\mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^{\top}$ propagates the previous uncertainty through the dynamics and shows how errors in the old state are transformed by the motion model. The matrix $\mathbf{Q}_{k-1}$ then adds uncertainty due to the process noise. In this way, the covariance prediction step increases or reshapes the uncertainty to reflect both the effect of the dynamics and the modeling errors between time steps.

3.7.3 Kalman Gain

The Kalman gain $\mathbf{K}_k$ controls how much the filter corrects the prediction using the new measurement. It is given by [22],

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^\top \left(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^\top + \mathbf{R}_k\right)^{-1}, \tag{3.12}$$

where $\mathbf{P}_{k|k-1}$ is the predicted state covariance, $\mathbf{H}_k$ is the measurement matrix, and $\mathbf{R}_k$ is the measurement noise covariance. The numerator $\mathbf{P}_{k|k-1}\mathbf{H}_k^\top$ describes how uncertainty in the state appears in measurement space. The term in parentheses,

$$\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^\top + \mathbf{R}_k,$$

is the innovation covariance, which combines uncertainty from the predicted state and the sensor noise. When $\mathbf{R}_k$ is large, the innovation covariance is large and the gain $\mathbf{K}_k$ becomes small, so the filter trusts the prediction more than the measurement. When $\mathbf{R}_k$ is small, the innovation covariance is smaller and the gain increases, so the filter gives more weight to the new measurement in the update.

3.7.4 State Update

The updated state estimate is computed as [22],

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}\right), \tag{3.13}$$

where $\mathbf{K}_k$ is the Kalman gain and

$$\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}$$

is the measurement residual. The measurement residual is the difference between the measurement and the predicted measurement from the current state estimate. The Kalman gain scales the residual to correct the predicted state and controls how strongly the measurement changes the updated estimate.

3.7.5 Covariance Update

After incorporating the measurements at time k, the posterior covariance is [22],

$$\mathbf{P}_{k|k} = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k\right) \mathbf{P}_{k|k-1}, \tag{3.14}$$

where $\mathbf{P}_{k|k-1}$ is the predicted covariance, $\mathbf{K}_k$ is the Kalman gain, and $\mathbf{I}$ is the identity matrix. The matrix $(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)$ reduces the prior uncertainty. State dimensions that are well observed by the sensor, and therefore have a large effective Kalman gain, experience a larger decrease in variance. State dimensions that are weakly observed have a low Kalman gain, so their uncertainty remains close to the predicted value.

3.8 Extended Kalman Filter (EKF)

The Kalman filter assumes that the state transition and measurement models are linear. However, many real world systems are nonlinear. The Extended Kalman Filter enables the use of the Kalman Filter with nonlinear models by local linearization.

3.8.1 Linearization

Using first-order Taylor expansion about $\hat{\mathbf{x}}_{k|k-1}$:

$$f_k(\mathbf{x}_k) \approx f_k(\hat{\mathbf{x}}_{k|k-1}) + \mathbf{F}_k(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}), \tag{3.15}$$

$$h_k(\mathbf{x}_k) \approx h_k(\hat{\mathbf{x}}_{k|k-1}) + \mathbf{H}_k(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}), \tag{3.16}$$

where $\mathbf{F}_k = \frac{\partial f_k}{\partial \mathbf{x}}|_{\hat{\mathbf{x}}_{k|k-1}}$ and $\mathbf{H}_k = \frac{\partial h_k}{\partial \mathbf{x}}|_{\hat{\mathbf{x}}_{k|k-1}}$.

3.8.2 Nonlinear System Model

$$\mathbf{x}_{k+1} = f_k(\mathbf{x}_k, \mathbf{v}_k), \tag{3.17}$$

$$\mathbf{z}_k = h_k(\mathbf{x}_k, \mathbf{w}_k), \tag{3.18}$$

where $f_k(\cdot)$ and $h_k(\cdot)$ are nonlinear functions, and $\mathbf{v}_k$ and $\mathbf{w}_k$ are zero-mean Gaussian noises with covariances $\mathbf{Q}_k$ and $\mathbf{R}_k$ [22],.

3.8.3 EKF Prediction and Update

Similarly to the Kalman Filter, the Extended Kalman Filter (EKF) consists of two main steps: prediction and update. In the prediction step, the state estimate and covariance are propagated forward using the nonlinear process model [22]:

$$\hat{\mathbf{x}}_{k|k-1} = f_k(\hat{\mathbf{x}}_{k-1|k-1}), \tag{3.19}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^\top + \mathbf{Q}_k. \tag{3.20}$$

The update step incorporates the new measurement to correct the predicted estimate [22]:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^\top \left(\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^\top + \mathbf{R}_k \right)^{-1}, \tag{3.21}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \left[\mathbf{z}_k - h_k(\hat{\mathbf{x}}_{k|k-1}) \right], \tag{3.22}$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}. \tag{3.23}$$

The EKF provides a first-order approximation of nonlinear dynamics. Its performance depends on the validity of the linearization and noise assumptions.

3.8.4 Jacobian for Radar Measurement Models

In maritime radar applications, nonlinearity arises from the coordinate mismatch between the sensor and the tracker. Radar natively provides observations in polar coordinates (range r and bearing θ), while target states are typically tracked in Cartesian coordinates $[x, y, \dot{x}, \dot{y}]^\top$. This transformation is defined by the nonlinear measurement function $h(\mathbf{x})$:

$$\mathbf{z} = h(\mathbf{x}) = \begin{bmatrix} \sqrt{x^2 + y^2} \\ \tan^{-1}(y/x) \end{bmatrix}. \tag{3.24}$$

To utilize the EKF, the measurement Jacobian matrix $\mathbf{H}_k$ must be computed by differentiating $h(\mathbf{x})$ with respect to the state variables:

$$\mathbf{H}_k = \begin{bmatrix} \frac{x}{\sqrt{x^2+y^2}} & \frac{y}{\sqrt{x^2+y^2}} & 0 & 0 \\ \frac{-y}{x^2+y^2} & \frac{x}{x^2+y^2} & 0 & 0 \end{bmatrix}. \tag{3.25}$$

This matrix transforms the cartesian state uncertainty into the polar space to enable the prediction and update steps.

3.9 Markov Chains

A Markov Chain is a discrete-time stochastic process

$$\{X_k\}_{k \geq 0}, \quad X_k \in \mathcal{X},$$

defined on a state space $\mathcal{X}$. The defining feature of a Markov Chain is the Markov property, which states that the conditional distribution of the next state depends only on the present state and not on the full history of the process [23]:

$$\mathbb{P}(X_{k+1} \mid X_k, X_{k-1}, \ldots, X_0) = \mathbb{P}(X_{k+1} \mid X_k).$$

This expresses that the present state contains all the information needed to determine the future evolution. In the context of state estimation, the Markov assumption implies that the current state X_k contains all relevant information from the past. As a result, recursive filters only need the most recent estimate and the current measurement to perform prediction and update steps, rather than the full history of the process.

Chapter 4

Multi-Object Tracking

4.1 Introduction

In practical radar systems, targets return multiple measurements and scenarios often contain clutter measurements. Single-target filters such as the Kalman or Extended Kalman Filter assume there is one measurement per target, which becomes invalid when the number of targets is unknown and time-varying.

The *multi-object tracking* (MOT) problem seeks to estimate both how many targets exist and where they are, given radar measurements that may include false alarms and missed detections. To formulate this problem rigorously, we extend the single-target Bayesian filter to operate on *sets* of states and measurements rather than on individual vectors. The *Random Finite Set* (RFS) framework forms the foundation of the Probability Hypothesis Density (PHD) Filter.

4.2 Random Finite Set Representation

Let

$$X_k = \{x_1, x_2, \ldots, x_{M(k)}\} \tag{4.1}$$

denote the set of all target states at time k, and

$$Z_k = \{z_1, z_2, \ldots, z_{N(k)}\} \tag{4.2}$$

denote the corresponding measurement set. Unlike vector representations, Random Finite Sets (RFS) treat both the number of elements and their values as random variables. This means that the order and amount of elements is irrelevant and removes the need for explicit data association, which is one of the primary computational burdens of traditional multi-object tracking. The multi-object state X_k is an RFS on a state space $\mathcal{X} \subset \mathbb{R}^n$, described by:

- A probability distribution that tells how likely it is to have $0, 1, 2, \ldots$ objects at time k (the *cardinality* $|X_k|$), and
- For each possible number of objects, a probability density that describes how likely different configurations of their states are, treating the objects as unlabeled and interchangeable.

Thus, uncertainty is represented not only in the state values but also in the number of objects present.

4.3 Multi-Object Bayesian Filtering

Extending the recursive Bayesian formulation from Chapter 3, the multi-object Bayes filter [24] propagates the posterior density $p_k(X_k|Z_{1:k})$ over finite sets:

$$p_{k|k-1}(X_k|Z_{1:k-1}) = \int f_{k|k-1}(X_k|X_{k-1})\, p_{k-1}(X_{k-1}|Z_{1:k-1})\, \mu_s(dX), \tag{4.3}$$

$$p_k(X_k|Z_{1:k}) = \frac{g_k(Z_k|X_k)\, p_{k|k-1}(X_k|Z_{1:k-1})}{\int g_k(Z_k|X)\, p_{k|k-1}(X|Z_{1:k-1})\, \mu_s(dX)}, \tag{4.4}$$

where $f_{k|k-1}(X_k|X)$ is the *multi-object transition density*, $g_k(Z_k|X_k)$ is the *multi-object likelihood*, and μ_S denotes the reference measure on the space of finite sets. These equations are the set-valued analogues of the single-target prediction and update steps. In theory, they yield the optimal solution to the multi-object filtering problem.

4.4 The Challenge of Exact Multi-Object Filtering

Due to the multi-object Bayes filter being an exact calculation of the posterior distribution, it is *computationally intractable* in most real scenarios. At each time step, it must integrate over all possible combinations of surviving, born, and undetected targets and all possible associations between measurements and targets. This means that it is unpractical to compute the posterior distribution with realistic time constraints.

If there are n existing targets and m new measurements, the number of possible association hypotheses is

$$N(n,m) = \sum_{k=0}^{\min(n,m)} \frac{n!\,m!}{(n-k)!\,(m-k)!\,k!}, \tag{4.5}$$

which grows factorially with both n and m. Even conservative values such as $n = m = 10$ where there are 10 detections and 10 targets produces tens of millions hypotheses. Therefore, exact computation of the set integrals is not a practical approach to multi-object tracking. This motivates the need for approximate formulations.

4.5 The Probability Hypothesis Density

To overcome this computational problem, we can approximate the posterior distribution with its *first-order statistical moment*, known as the *Probability Hypothesis*

Density (PHD). For a random finite set Σ defined on the state space $\mathcal{X}$, the first-order moment or *intensity function* $v_{\sum}(x)$ is a nonnegative function satisfying

$$\mathbb{E}[|\Sigma \cap S|] = \int_S v_{\sum}(x)\, dx, \tag{4.6}$$

for any region $S \subseteq \mathcal{X}$, where $|\Sigma \cap S|$ denotes the number of elements of Σ within S. In this interpretation, $v_{\sum}(x)$ represents the expected number of targets per unit volume at location x, and integrating it over some region yields the expected number of targets. [25].

4.6 From Multi-Object Bayes Filter to the PHD Approximation

Mahler proposed to propagate only this first-order moment through time as opposed to the posterior distribution [24]. By integrating the multi-object Bayes prediction and update equations over all but one target variable, one obtains recursive equations for $v_k(x)$ that approximate the expected multi-object behavior without data associations. This yields the *PHD filter*, which captures the properties of the multi-object system while maintaining linear growth with the number of measurements rather than factorial growth. The following chapter defines these recursive equations and develops the *Gaussian Mixture PHD (GM-PHD)* filter, a closed-form implementation suitable for radar-based multi-object tracking [3].

Chapter 5

The Probability Hypothesis Density (PHD) Tracker

In the previous chapter, the Random Finite Set (RFS) framework was introduced as a rigorous means of representing multi-object systems where both the number of targets and their individual states are uncertain.

To obtain a tractable approximation, this chapter develops the *Probability Hypothesis Density* (PHD) filter, which propagates only the first-order statistical moment (the intensity) of the multi-object posterior. The following sections define the recursive PHD equations and the closed-form implementation with Gaussian mixtures. Note this section is based on the formulation presented in Vo and Ma's paper [3].

5.1 Multi-Object Sets

The PHD filter is based on a Random Finite Set (RFS) model for the temporal evolution of the multi-target state, accounting for target motion, survival, birth, and death [3]:

$$X_k = \left(\bigcup_{\zeta \in X_{k-1}} S_{k|k-1}(\zeta) \right) \cup \left(\bigcup_{\zeta \in X_{k-1}} B_{k|k-1}(\zeta) \right) \cup \Gamma_k. \tag{5.1}$$

For every previous target state $\zeta \in X_{k-1}$, the set $S_{k|k-1}(\zeta)$ describes the states that survive to time k according to the motion model and the probability of survival $p_{S,k}(\zeta)$. This corresponds to the prediction of existing targets in the PHD filter. The set $B_{k|k-1}(\zeta)$ describes new targets that are spawned from the existing state ζ, while Γ_k represents spontaneously born targets that do not come from any previous state.

At time k, each target $x \in X_k$ is detected with probability $p_{D,k}(x)$ and produces a measurement distributed according to $g_k(z|x)$. The sensor also receives a Poisson-distributed clutter RFS K_k with intensity $\kappa_k(z)$. Hence, the measurement set is [3]

$$Z_k = K_k \cup \left(\bigcup_{x \in X_k} \Theta_k(x) \right), \tag{5.2}$$

where $\Theta_k(x)$ is either $\{z\}$ if the target is detected or $\emptyset$ if it is missed.

5.2 PHD Filter Equations

The PHD filter recursively predicts and updates the intensity function $v_k(x)$.

5.2.1 Prediction

The prediction step is based on the superposition of two PHDs [3],

$$v_{k|k-1}(x) = \int p_{S,k}(\zeta) f_{k|k-1}(x|\zeta) v_{k-1}(\zeta)\, d\zeta + \int \beta_{k|k-1}(x|\zeta) v_{k-1}(\zeta)\, d\zeta + \gamma_k(x). \tag{5.3}$$

In this first term, the mass $v_{k-1}(\zeta)$ is propagated forward through the motion model $f_{k|k-1}(x|\zeta)$ and is scaled by the probability of survival $p_{S,k}(\zeta)$. This describes how the previous targets move according to the motion model and scaled by a factor that accounts for the possibility that some of these targets will disappear.

The second terms models how new targets are spawned from existing ones. Starting with the previous targets $v_{k-1}(\zeta)$, the spawn model $\beta_{k|k-1}(x|\zeta)$ adds intensity around those states. This represents the possibility of new targets generating from the previous targets.

5.2.2 Update

Given the measurement set Z_k, the updated PHD is [3],

$$v_k(x) = \left[1 - p_{D,k}(x)\right] v_{k|k-1}(x) + \sum_{z \in Z_k} \frac{p_{D,k}(x)\, g_k(z|x)\, v_{k|k-1}(x)}{\kappa_k(z) + \int p_{D,k}(\xi)\, g_k(z|\xi)\, v_{k|k-1}(\xi)\, d\xi}. \tag{5.4}$$

The first term keeps the part of the predicted intensity $v_{k|k-1}(x)$ that corresponds to targets that were not detected. The factor $\left[1 - p_{D,k}(x)\right]$ is the probability that a real target at state x has no measurement at time k. This allows the filter to slowly decay targets that are not detected, while still keeping some probability that they may be detected again at a future time step. The second term of the numerator,

$$p_{D,k}(x)\, g_k(z|x)\, v_{k|k-1}(x),$$

describes how likely a particular measurement z given that it comes from state x. The factor $p_{D,k}(x)$ is the probability that a target at x is detected and $g_k(z|x)$ is the likelihood of observing measurement z if the target is at x. The numerator is normalized by the clutter rate and the contributions from all other possible target states. This ensures that the total contribution of measurement z is spreads across the entire state space.

5.3 Gaussian Mixture PHD Implementation

To obtain a closed-form realization, Vo and Ma [3] approximate the PHD by a Gaussian mixture:

$$v_k(x) = \sum_{i=1}^{J_k} w_k^{(i)}\, \mathcal{N}(x; m_k^{(i)}, P_k^{(i)}), \tag{5.5}$$

and assume linear–Gaussian motion and measurement models, Poisson clutter, and state-independent survival/detection probabilities.

5.3.1 Prediction Step

The predicted PHD intensity is then the superposition of all surviving target components and the birth intensity [3]:

$$v_{k|k-1}(x) = v_{S,k|k-1}(x) + v_{\beta,k|k-1}(x) + \gamma_k(x) \tag{5.6}$$

where the posterior Gaussian mixture at time $k-1$ for surviving components is,

$$v_{S,k|k-1}(x) = p_{S,k} \sum_{j=1}^{J_{k-1}} w_{k-1}^{(j)} \mathcal{N}(x; m_{S,k|k-1}^{(j)}, P_{S,k|k-1}^{(j)}) \tag{5.7}$$

The survival predicted mixture is obtained by applying the motion model to each survival and birth component's mean, covariance, and weight [3]:

$$m_{S,k|k-1}^{(j)} = F_{k-1}\, m_{k-1}^{(j)}, \tag{5.8}$$

$$P_{S,k|k-1}^{(j)} = F_{k-1}\, P_{k-1}^{(j)}\, F_{k-1}^{\top} + Q_{k-1} \tag{5.9}$$

The birth Gaussian mixture at time $k-1$ is,

$$v_{\beta,k|k-1}(x) = \sum_{j=1}^{J_{k-1}} \sum_{l=1}^{J_{\beta,k}} w_{k-1}^{(j)} w_{\beta,k}^{(l)} \mathcal{N}(x; m_{\beta,k}^{(j,l)}, P_{\beta,k}^{(j,l)}) \tag{5.10}$$

This birth Gaussian mixture describes how targets are spawned from the existing J_{k-1} components. The component's mean, covariance and weight are calculated by [3],

$$m_{\beta,k|k-1}^{(j,l)} = F_{\beta,k-1}^{(l)} m_{k-1}^{(j)} + d_{\beta,k-1}^{(l)}, \tag{5.11}$$

$$P_{\beta,k|k-1}^{(j,l)} = Q_{\beta,k-1}^{(l)} + F_{\beta,k-1}^{(l)} P_{k-1}^{(j)} (F_{\beta,k-1}^{(l)})^T \tag{5.12}$$

Overall, the prediction step estimates the posterior intensity of the target set by incorporating two key processes: the Kalman-like motion model describing the surviving Gaussian components, and the Poisson birth process accounting for the emergence and spawning of new targets.

5.3.2 Update Step

For each measurement $z \in Z_k$, each predicted Gaussian component generates a new component according to the Kalman update equations. This process corrects the predicted state estimate based on the measurement likelihood and the detection probability, effectively redistributing the mixture weights toward components that are consistent with the observed data. The *Kalman gain* determines how strongly the new measurement influences the updated estimate [3]:

$$K_k^{(j)} = P_{k|k-1}^{(j)} H_k^\top \left(H_k P_{k|k-1}^{(j)} H_k^\top + R_k \right)^{-1}. \tag{5.13}$$

The mean and covariance of each component are then corrected using this gain. The updated mean shifts toward the measurement, while the covariance contracts to reflect the reduced uncertainty after the update:

$$m_k^{(j,z)} = m_{k|k-1}^{(j)} + K_k^{(j)} \left(z - H_k m_{k|k-1}^{(j)} \right), \tag{5.14}$$

$$P_k^{(j,z)} = \left(I - K_k^{(j)} H_k \right) P_{k|k-1}^{(j)}. \tag{5.15}$$

Each updated component receives a new weight proportional to both its prior weight and the likelihood of producing the measurement z:

$$\tilde{w}_k^{(j,z)} = \frac{p_{D,k}\, w_{k|k-1}^{(j)}\, \mathcal{N}\left(z; H_k m_{k|k-1}^{(j)}, S_k^{(j)} \right)}{\kappa_k(z) + \sum_{i=1}^{J_{k|k-1}} p_{D,k}\, w_{k|k-1}^{(i)}\, \mathcal{N}\left(z; H_k m_{k|k-1}^{(i)}, S_k^{(i)} \right)}. \tag{5.16}$$

The denominator acts as a normalization factor that accounts for clutter and competing hypotheses from all other predicted components.

If a component is not associated with any measurement, it is retained with a reduced weight proportional to the probability of missed detection, while its mean and covariance remain unchanged:

$$w_k^{(j)} = (1 - p_{D,k})\, w_{k|k-1}^{(j)}, \tag{5.17}$$

$$m_k^{(j)} = m_{k|k-1}^{(j)}, \tag{5.18}$$

$$P_k^{(j)} = P_{k|k-1}^{(j)}. \tag{5.19}$$

Together, these updates shift the PHD intensity toward regions of the state space that are consistent with the observed measurements, while suppressing unlikely hypotheses. Each Gaussian component now represents an updated target hypothesis conditioned on either detection or missed detection.

5.3.3 Merging and Pruning of Gaussian Components

In the Gaussian Mixture PHD (GM-PHD) filter, the posterior intensity at each time step is represented as a weighted sum of Gaussian components:

$$v_k(x) = \sum_{i=1}^{J_k} w_k^{(i)} \mathcal{N}\left(x; m_k^{(i)}, P_k^{(i)}\right), \tag{5.20}$$

where $w_k^{(i)}$ denotes the weight, $m_k^{(i)}$ the mean, and $P_k^{(i)}$ the covariance of the i-th component. Over time, the number of components J_k increases due to repeated prediction, birth, and measurement update steps. Without control, this exponential growth can lead to excessive computational cost and redundant mixture components that represent nearly identical hypotheses.

To maintain tractability, the GM-PHD filter applies two reduction procedures at the end of each update cycle: *pruning* and *merging*. Pruning removes components whose weights fall below a predefined threshold T, thereby discarding weak hypotheses that contribute negligibly to the overall intensity. Merging then combines components that are statistically similar to prevent redundant representation of the same target.

The merging procedure begins by selecting the component with the largest weight and identifying all components whose Mahalanobis distance from it is below a user-specified threshold U. These components are grouped into a cluster and replaced by a single equivalent Gaussian. Let $\mathcal{L}$ denote the set of indices of components to be merged. The merged component parameters are computed as [3],

$$\tilde{w}_k = \sum_{i \in \mathcal{L}} w_k^{(i)}, \tag{5.21}$$

$$\tilde{m}_k = \frac{1}{\tilde{w}_k} \sum_{i \in \mathcal{L}} w_k^{(i)} m_k^{(i)}, \tag{5.22}$$

$$\tilde{P}_k = \frac{1}{\tilde{w}_k} \sum_{i \in \mathcal{L}} w_k^{(i)} \left[P_k^{(i)} + \left(m_k^{(i)} - \tilde{m}_k\right)\left(m_k^{(i)} - \tilde{m}_k\right)^\top \right]. \tag{5.23}$$

This process is repeated until all remaining components have been considered. If the total number of components after merging exceeds a specified upper limit $J_{\max}$, only the $J_{\max}$ components with the largest weights are retained. The resulting reduced set forms the posterior intensity used for the next iteration. Through these steps, the GM-PHD filter maintains a balance between representational richness and computational efficiency.

5.3.4 Birth and Spawn Modeling

New targets are introduced into the Gaussian Mixture PHD framework through a Poisson Random Finite Set (RFS) that models both spontaneous target births and targets spawned from existing ones. The birth intensity specifies regions in the state space where new targets are expected to appear, while the spawn intensity describes new targets generated from current tracks.

At each time step, the predicted intensity includes contributions from surviving, spawned, and spontaneously born targets. The spontaneous birth intensity is typically modeled as a Gaussian [3],

$$\gamma_k(x) = \sum_{i=1}^{J_{\gamma,k}} w_{\gamma,k}^{(i)} \mathcal{N}\left(x;\, m_{\gamma,k}^{(i)},\, P_{\gamma,k}^{(i)}\right), \tag{5.24}$$

where each Gaussian component defines a spatial region with nonzero probability of target birth. If no measurements subsequently confirm these hypotheses, their weights decay through the update process, effectively removing false births over time.

In addition to spontaneous births, the GM-PHD filter accounts for target spawning from existing objects through the spawn intensity

$$\beta_{k|k-1}(x \mid \zeta) = \sum_{j=1}^{J_{\beta,k}} w_{\beta,k}^{(j)} \mathcal{N}\left(x;\, F_{\beta,k-1}^{(j)}\zeta + d_{\beta,k-1}^{(j)},\, Q_{\beta,k-1}^{(j)}\right), \tag{5.25}$$

where each term represents a Gaussian component describing how a target with previous state ζ may generate new targets. The combined use of birth and spawn intensities allows the GM-PHD filter to handle both independent target appearances and target-induced spawns within a unified probabilistic framework. Hypothetical birth components are automatically pruned over time unless reinforced by consistent detections, maintaining the robustness of the overall model.

Chapter 6

Extended Target Modeling

In many formulations of multi-object tracking, the point-target assumption is commonly adopted. While this is convenient and forms the basis for classical tracking algorithms, it does not accurately reflect the characteristics targets' have in real world scenarios. A target in a maritime context can often return multiple measurements. This occurs because the target has a spatial extent, multiple surfaces for the radar signal to reflect. Therefore, the point-target assumption is insufficient because it fails to capture the possibility that multiple measurements correspond to one target.

To overcome this limitation, extended target models are employed. These models explicitly account for the possibility that each target may generate several measurements at each time step. One approach is to use probabilistic modeling, where the number and spatial distribution of measurements are described statistically, as in the random matrix or gamma-Gaussian inverse Wishart models. Another approach is to employ clustering-based processing, in which detections are grouped into measurement clusters assumed to correspond to individual targets.

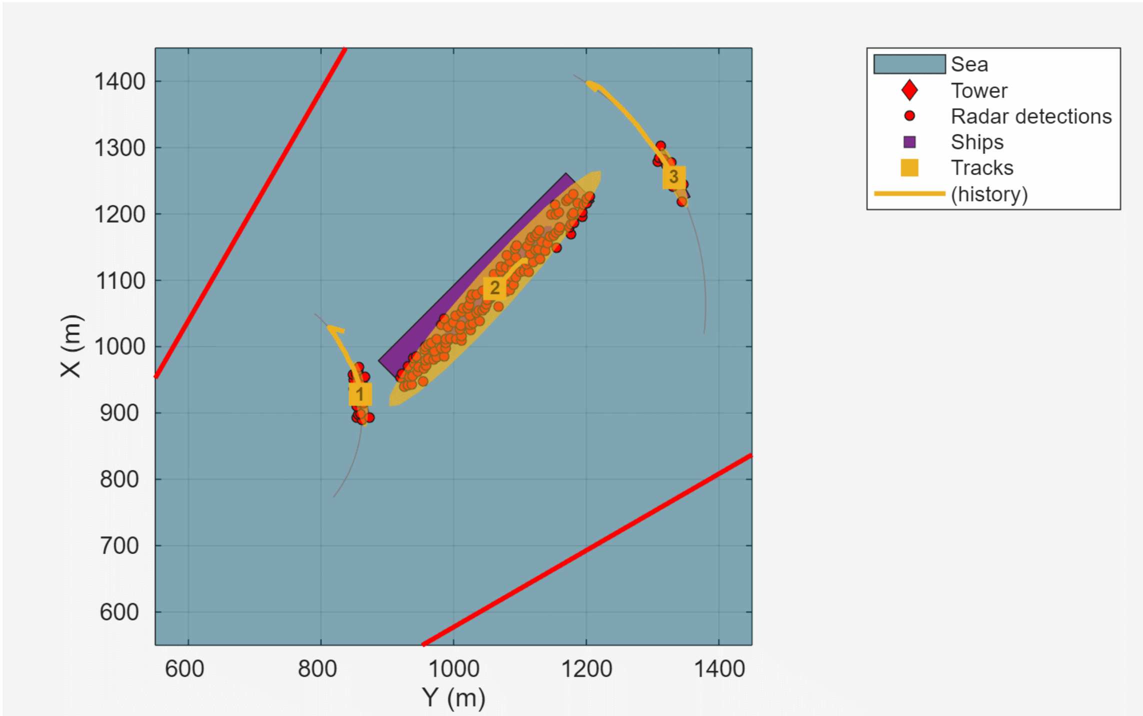

Figure 6.1: Illustration of possible extended targets in a MATLAB-simulated environment generated using synthetic radar parameters.

6.1 Wishart Distribution

The Wishart Distribution, introduced by John Wishart [26], is a multivariate generalization of the univariate chi-squared distribution. It is a probability distribution over symmetric positive semidefinite matrices, typically used to model covariance matrices. The diagonal elements of a Wishart-distributed matrix are chi-squared random variables, while the off-diagonal elements capture covariances between dimensions. Suppose X is a $p \times n$ matrix, each column of which is independently drawn from a p-variate normal distribution with zero mean:

$$X = (x_1, \ldots, x_n) \sim \mathcal{N}_p(0, \Sigma),$$

where each column

$$x_i = (x_{i,1}, \ldots, x_{i,p})^T \stackrel{iid}{\sim} \mathcal{N}_p(0, \Sigma), \quad \forall i \in \{1, \ldots, n\}.$$

Then, the $p \times p$ random matrix

$$S = XX^T = \sum_{i=1}^{n} x_i x_i^T$$

follows a Wishart distribution with scale matrix Σ and n degrees of freedom:

$$S \sim W_p(\Sigma, n).$$

The positive integer n acts as the degrees of freedom, analogous to the degrees of freedom parameter of a univariate chi-squared distribution. For $n \geq p$, the matrix S is invertible with probability 1 if Σ is invertible. When $p = 1$ and $\Sigma = 1$, the Wishart distribution reduces to a chi-squared distribution with n degrees of freedom.

Intuitively, just as the chi-squared distribution can be constructed by summing the squares of independent zero-mean univariate normal random variables, the Wishart distribution is obtained by summing the inner products of independent, zero-mean multivariate normal vectors. This property makes it a natural choice for modeling the distribution of the sample covariance matrix of multivariate normal data, after appropriate scaling by the sample size.

In the context of extended target tracking, the Wishart distribution underpins models of the scatter of measurements from a single extended target. Its conjugate prior, the Inverse Wishart distribution, is commonly used to represent uncertainty in the target's extent covariance, enabling recursive updates as new measurements are obtained.

6.2 Inverse Wishart Distribution

The Inverse Wishart distribution is the conjugate prior for the covariance matrix of a multivariate normal distribution. It is defined in terms of the Wishart distribution. A matrix $\mathbf{X}$ is said to follow an Inverse Wishart distribution, denoted as

$$\mathbf{X} \sim \mathcal{W}^{-1}(\mathbf{\Psi}, \nu),$$

if its inverse $\mathbf{X}^{-1}$ has a Wishart distribution with scale matrix $\mathbf{\Psi}^{-1}$ and ν degrees of freedom:

$$\mathbf{X}^{-1} \sim W_p(\mathbf{\Psi}^{-1}, \nu).$$

Here, $\mathbf{\Psi}$ is a symmetric positive definite scale matrix that determines the typical size of $\mathbf{X}$, and $\nu > p-1$ is the degrees of freedom, controlling how tightly the distribution is concentrated around $\mathbf{\Psi}$. Each eigenvalue of $\mathbf{X}$ behaves like an inverse-chi-squared random variable, and the distribution captures uncertainty about both the magnitude and orientation of a covariance matrix.

The Inverse Wishart distribution inherits many properties from the Wishart distribution. For example, suppose a matrix of measurements has a Wishart-distributed scatter matrix $S = XX^T$. In that case, the posterior distribution for the covariance (or extent) can be modeled as an Inverse Wishart. This makes it particularly useful in extended target tracking, where the extent covariance of a target can be updated recursively as new measurements are obtained:

$$\mathbf{\Sigma} \mid X_1, \ldots, X_n \sim \mathcal{W}^{-1}\left(\mathbf{\Psi} + \sum_{i=1}^{n} X_i X_i^T, \nu + n\right).$$

In this context, the Inverse Wishart provides a principled way to model uncertainty in the target's spatial extent, allowing the tracking algorithm to account for variability in the measurements while maintaining computational tractability.

6.3 Application of Extended Target Tracking

Extended target tracking methods aim to jointly estimate both the kinematic state and spatial extent of objects that generate multiple detections per scan. A widely used approach is the Gamma–Gaussian–Inverse–Wishart (GGIW) PHD filter, which models each target by coupling its kinematic state with an extent matrix that describes the spatial spread of its measurements [27]. However, the GGIW formulation relies on assumptions that are not always satisfied in maritime radar data. Specifically, it assumes that each target produces a dense and approximately elliptical cluster of detections whose spread is directly related to its physical extent. In practice, the maritime environment may have targets in which their measurement density does not align with the filter's elliptical cluster assumptions.

Therefore, this work adopts clustering and is used to approximate the grouping behavior assumed by extended-target models. HDBSCAN is applied to each radar frame to aggregate detections that likely originate from the same physical vessel. The resulting clusters are then used as inputs to the GM-PHD filter.

In summary, the assumptions underlying the GGIW-PHD filter motivate the use of a clustering-based preprocessing stage. By emulating the measurement-grouping behavior of extended-target models through HDBSCAN, the proposed pipeline achieves a balance between physical interpretability and real-time feasibility in radar-based maritime surveillance.

Chapter 7

Implementation and Simulation of the Tracking Algorithms

7.1 Overview

The previous chapters established the theoretical foundations for radar-centered multi object tracking and extended-target modeling. This chapter focuses on the experimental verification of the Gaussian Mixture Probability Hypothesis Density (GM-PHD) filter using real-world maritime radar measurements.

All experiments in this chapter are conducted using publicly available maritime radar datasets provided by the German Aerospace Center (DLR). These datasets contain radar measurements collected from both stationary and moving platforms, environmental clutter, and extended radar reflections. These conditions present challenges for point-target trackers, making them suited for assessing the robustness of the GM-PHD filter.

7.2 Dataset Description and Pre-Processing

7.2.1 Overview of Datasets

All experimental validations of the GM-PHD filter are performed using maritime radar datasets developed by the German Aerospace Center (DLR) and distributed under DLR export control conditions [1]. These datasets contain real radar measurements recorded from maritime routes in the Baltic Sea using an X-band marine radar operating at 1 Hz. Each dataset combines radar imagery with Automatic Identification System (AIS) reference tracks for multiple vessels observed under varying environmental conditions. Measurements are originally provided in polar coordinates (r_j, ψ_j) and transformed into the East–North–Up (ENU) Cartesian frame using AIS-derived own-ship positions. Effective radar resolutions are approximately 6 m per pixel in MANV and 11 m per pixel in both DAAN and DARC.

Table 7.1 summarizes the characteristics of the three DLR campaigns, and Figure 7.1 illustrates representative radar detections and corresponding AIS ground truth. Together, these datasets provide a benchmark for evaluating the GM-PHD filter across static and dynamic sensing configurations and a range of clutter conditions.

Table 7.1: Summary of DLR Marine Radar Datasets [1]

Dataset	**# Targets**	**# Frames**	**Duration (s)**	**Radar Platform**
MANV	6	976	1000	Static (observer)
DARC	2	527	540	Dynamic (own ship)
DAAN	4	780	780	Dynamic (own ship)

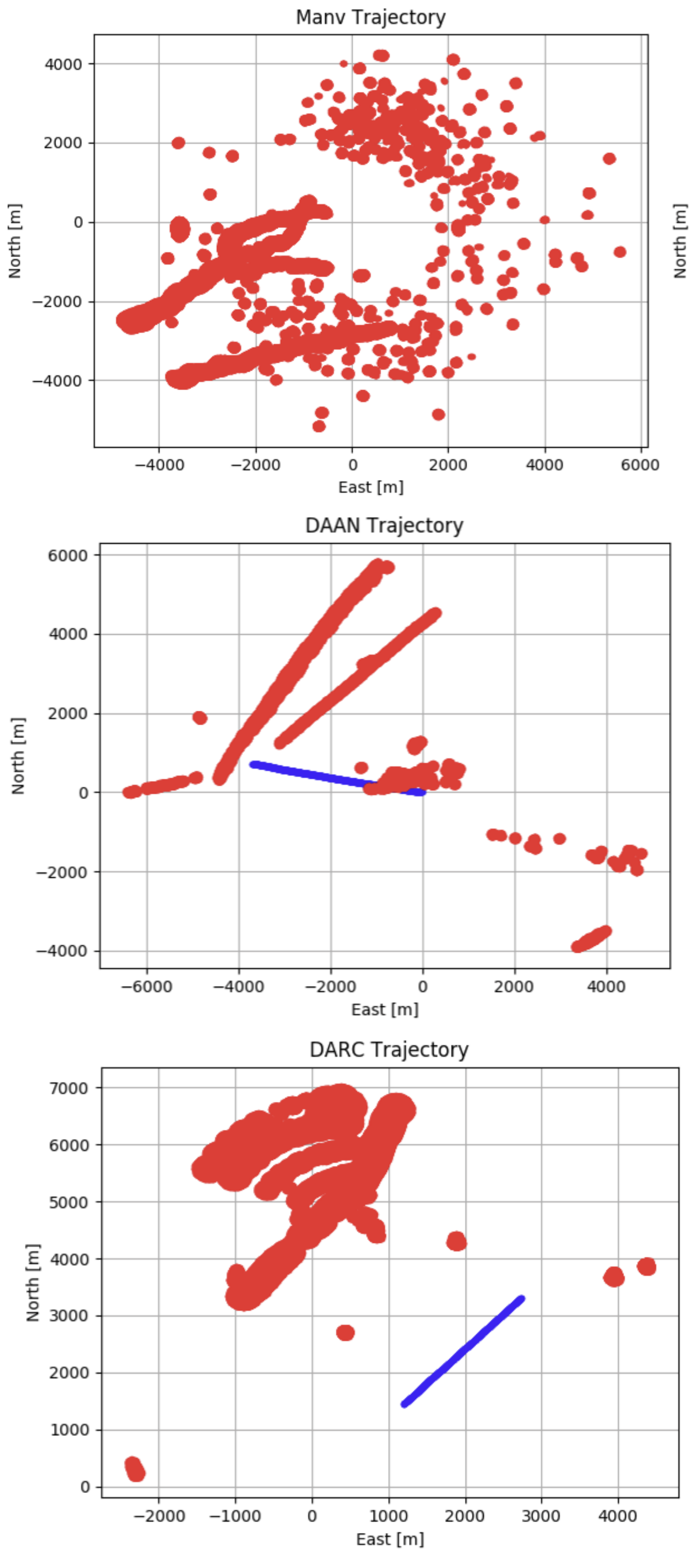

Figure 7.1: Representative radar detections and AIS trajectories for each dataset [1].

7.2.2 MANV Scenario

The MANV (Maneuvers) campaign provides a stationary-radar baseline for tracking evaluation. The radar is mounted on a fixed observation platform and observes six known targets over 1000 seconds, of which four are stationary and two perform maneuvering motion. Targets 2 and 3 correspond to moving vessels with approximate dimensions of 29×6.7 m and 23×6 m, respectively, exhibiting speed variations between 2 and 10 knots. Target 1 represents the static own ship. Residual sea-clutter returns and one uncooperative vessel without AIS transmission make this scenario a useful reference for evaluating the tracker's ability to handle clutter and unknown target identities.

7.2.3 DARC Scenario

The DARC (Data Association with RACON) campaign records radar data from a moving platform equipped with AIS over 540 seconds and includes two vessels. A unique feature of this dataset is the presence of a RACON beacon, a radar transponder that emits coded replies to incoming pulses. The RACON produces persistent reflections that overlap with target returns, especially in the first 400 frames. The dense clutter in this scenario provides a challenging scenario for the GM-PHD Filter.

7.2.4 DAAN

The DAAN (Data Association with Aids to Navigation) campaign captures 780 frames from a moving radar platform observing two vessels and several fixed navigational aids such as buoys and beacons. These static reflectors are persistent producing dense clutter. This scenario serves as another multi-object scenario with clutter to evaluate the tracker.

7.2.5 Data Pre-Processing Pipeline

The DLR Marine Radar Dataset provides detections that have already been extracted from raw radar images using the processing pipeline described in [1]. Their pipeline includes masking of radar displays, grayscale conversion, intensity thresholding, automatic blob detection using the Determinant of Hessians (DoH) method, and conversion from polar to Cartesian ENU coordinates. All of these steps are performed by DLR as part of the dataset release and are not reimplemented in this work. In this thesis, the only additional pre-processing step applied to the DLR-provided detections is the clustering stage:

1. **Clustering:** The DLR-provided point detections are grouped using the Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) algorithm. HDBSCAN is used in place of the k-means clustering, as it does not require specifying the number of clusters in advance and is more robust to irregular extended-target shapes and non-uniform measurement densities.

The resulting clustered detections $\mathcal{Z}_k$ form the measurement input to the GM-PHD filter.

7.3 Target and Measurement Models

The Gaussian Mixture Probability Hypothesis Density (GM-PHD) filter requires definitions of the target motion and measurement models, which describe how each tracked object evolves over time and how it is observed by the radar sensor. In this implementation, the target dynamics are modeled using a linear constant-velocity (CV) model, while two distinct measurement models are evaluated: a linear Cartesian model and a nonlinear bearing–range model. The DLR Marine Radar Dataset provides detections in both coordinate systems—raw radar detections in polar form (bearing and range) and preprocessed detections transformed into Cartesian East and North coordinates using the own-ship's AIS position and heading. This enables the comparison between the standard linear Kalman-based GM-PHD filter and the Extended Kalman GM-PHD filter (EKF-GM-PHD).

7.3.1 Linear Motion Model

Each target is represented by a four-dimensional state vector describing its planar position and velocity in Cartesian East–North coordinates:

$$\mathbf{x}_k = \begin{bmatrix} x_k & y_k & \dot{x}_k & \dot{y}_k \end{bmatrix}^T .$$

The state evolves according to the constant-velocity model

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + \mathbf{w}_k,$$

where $\mathbf{w}_k \sim \mathcal{N}(0, Q)$ is zero-mean Gaussian process noise, and the state transition matrix F is

$$F = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} .$$

Here, $\Delta T = 1$ s corresponds to the radar update rate in the DLR datasets. The model assumes linear motion between scans, with $\mathbf{w}_k$ capturing unmodeled dynamics.

7.3.2 Linear Cartesian Measurement Model

For detections that have been transformed into Cartesian coordinates, a linear measurement model is used:

$$\mathbf{z}_k = H\mathbf{x}_k + \mathbf{v}_k, \qquad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

where $\mathbf{v}_k \sim \mathcal{N}(0, R_k)$ is zero-mean Gaussian measurement noise with covariance

$$R_k = \begin{bmatrix} 0.75 & 0 \\ 0 & 0.75 \end{bmatrix}.$$

This configuration corresponds to the `LinearGaussian` measurement model in the Stone Soup framework. It assumes isotropic noise in the x and y directions and models radar detections that have already been converted from their native polar representation.

7.3.3 Nonlinear Bearing–Range Measurement Model

For raw polar detections, a nonlinear measurement model is employed that operates directly in the radar's native coordinate space. The measurement function is defined as

$$h(\mathbf{x}_k) = \begin{bmatrix} \tan^{-1}\left(\frac{y_k}{x_k}\right) \\ \sqrt{x_k^2 + y_k^2} \end{bmatrix},$$

and the corresponding measurement equation is

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k, \qquad \mathbf{v}_k \sim \mathcal{N}(0, R_k),$$

with covariance

$$R_k = \begin{bmatrix} \sigma_\theta^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix},$$

where $\sigma_\theta = 1^\circ$ and $\sigma_r = 10\,\mathrm{m}$. This configuration is used in the `CartesianToBearing` measurement model and is used in the Extended Kalman Filter implementation of the GM-PHD filter.

7.3.4 Discussion

The two configurations represent different but complementary uses of the DLR dataset. The linear Cartesian model operates on preprocessed detections, offering a fast and straightforward baseline for GM-PHD tracking. However, it assumes uniform uncertainty across both spatial axes, neglecting the directional nature of radar noise. The nonlinear bearing–range model, in contrast, directly utilizes the raw radar detections and preserves the true polar measurement geometry, in which range is measured with much higher fidelity than bearing. The performance of these two approaches is discussed in Chapter 8.

7.4 Filter Parameters and Control Mechanisms

In practice, the performance and stability of the Gaussian Mixture Probability Hypothesis Density (GM-PHD) filter depend strongly on several tunable parameters that control mixture reduction, gating, and clutter modeling. These parameters determine how aggressively the filter merges, discards, or weights information, and they must be selected to balance tracking accuracy with computational tractability. Table 7.2 summarizes the key control parameters used in this work, while the following subsections describe their functional roles and rationale for selection.

Table 7.2: Summary of primary GM-PHD filter control parameters.

Parameter	Symbol	Description
Merging threshold	T_m	Minimum Mahalanobis distance used to merge components.
Pruning threshold	T_p	Minimum Gaussian component weight retained in the mixture.
Missed-distance threshold	T_d	Maximum Mahalanobis distance for measurement association.
Clutter spatial density	λ_c	Uniform false-alarm (clutter) spatial density.
Detection probability	$p_{D,k}$	Probability that a target generates a valid detection.
Survival probability	$p_{S,k}$	Probability that a target persists to the next time step.

7.4.1 Gaussian Mixture Merging and Pruning

The GM-PHD filter represents its intensity as a Gaussian mixture, which can grow rapidly with each iteration. To prevent exponential expansion, the filter uses two complementary parameters—T_m for *merging* and T_p for *pruning*. The merging threshold T_m determines the Mahalanobis distance below which similar components are combined:

$$d_M(\mu_1, \mu_2) = \sqrt{(\mu_1 - \mu_2)^T P^{-1} (\mu_1 - \mu_2)}, \quad d_M < T_m.$$

Components that fall within this distance are merged into a single equivalent Gaussian, preserving the overall intensity while reducing redundancy. The pruning threshold T_p removes components whose normalized weights fall below a specified limit, typically indicating negligible contribution to the target density:

$$w_i < T_p.$$

For the DLR datasets, $T_p = 10^{-5}$ was found sufficient to eliminate spurious components caused by transient clutter while maintaining all significant targets. Both merging and pruning are implemented in Stone Soup using the `GaussianMixtureReducer` class, which automates the reduction step after each GM-PHD update.

7.4.2 Gating via Missed-Distance Threshold

Although the GM-PHD framework avoids explicit data association, a gating mechanism is used to exclude improbable measurement updates. The gating threshold T_d defines the maximum allowable Mahalanobis distance between a measurement z and a component mean μ [28]:

$$d_M(z, \mu) = \sqrt{(z - H\mu)^T (HPH^T + R)^{-1} (z - H\mu)}, \quad d_M < T_d.$$

This effectively filters out unlikely detections before the update step, reducing computational cost and minimizing false associations from sea clutter. This mechanism is realized in Stone Soup through the `DistanceHypothesiser` class with a Mahalanobis metric.

7.4.3 Clutter Spatial Density

The clutter intensity parameter λ_c models the expected density of false alarms in the measurement space. Recall that within the GM-PHD update, the denominator term $\kappa_k(z) = \lambda_c$ acts as a normalization factor between target-generated and clutter-generated detections:

$$D_{k|k}(x) = (1 - p_{D,k}) D_{k|k-1}(x) + \sum_{z \in Z_k} \frac{p_{D,k} g_k(z|x) D_{k|k-1}(x)}{\lambda_c + \int p_{D,k} g_k(z|\xi) D_{k|k-1}(\xi) d\xi}.$$

Larger λ_C values reduce the weight of new detections, effectively modeling denser clutter environments. This parameter is configured through the `PHDUpdater` class in Stone Soup.

7.4.4 Target Survival and Detection Probabilities

The probabilities of survival ($p_{S,k}$) and detection ($p_{D,k}$) determine how the GM-PHD filter balances persistence against spurious track generation. A high survival probability ($p_{S,k} = 0.9$) ensures that genuine tracks remain active through temporary missed detections, while a moderate detection probability ($p_{D,k} = 0.6$) accounts for missed detections and sea clutter. A moderate detection probability was chosen due to detection fragmentation, which allows the track to survive until another detection updates the track.

7.4.5 Discussion

The parameters described above collectively govern the stability and sensitivity of the GM-PHD filter. Merging and pruning thresholds (T_m, T_p) control the number of Gaussian components and computational complexity. The gating threshold (T_d) directly affects measurement association efficiency, while the clutter density λ_C determines the filter's tolerance to false alarms. Finally, the survival and detection probabilities ($p_{S,k}$, $p_{D,k}$) influence track continuity and robustness to missed detections. The values provided here were chosen based on trial and error for the DLR dataset to maximize performance.

7.5 Implementation Framework

The Gaussian Mixture Probability Hypothesis Density (GM-PHD) tracker and its evaluation framework were implemented using the open-source *Stone Soup* library [2].

Stone Soup provides a modular and extensible architecture for multi-object tracking and state estimation, enabling the integration of customized prediction, update, and hypothesis management components. All experiments in this thesis were developed in Python using Stone Soup's GM-PHD filter base classes, with additional modules implemented for radar-specific preprocessing and extended target handling.

7.5.1 Framework Architecture

The Stone Soup framework organizes tracking into modular stages consisting of prediction, update, hypothesis generation, and mixture reduction. This modularity allowed for the integration of both the linear Kalman-based GM-PHD filter and the nonlinear Extended Kalman GM-PHD (EKF-GM-PHD) variant, corresponding to the measurement models discussed previously. Figure 7.2 illustrates the overall data flow structure, where radar detections are ingested, filtered through the GM-PHD recursion, and evaluated using performance metrics.

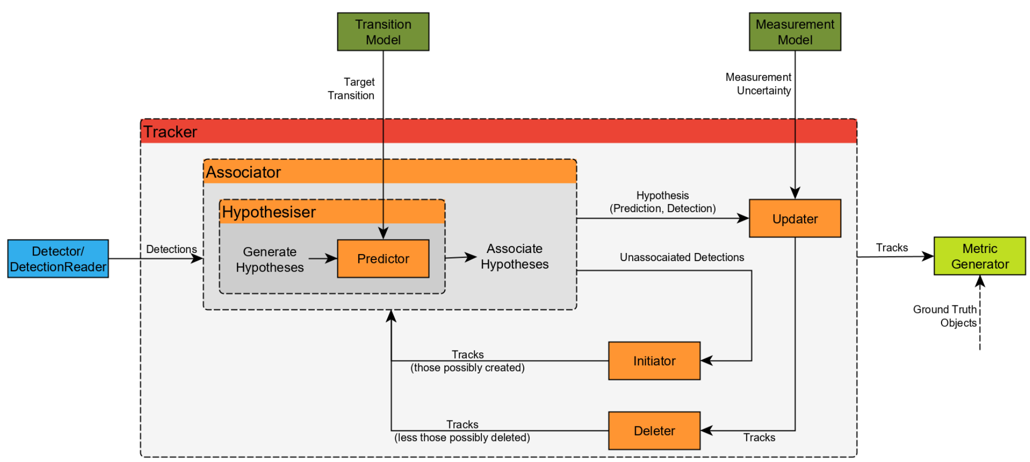

Figure 7.2: Modular data flow of the Stone Soup multi-object tracking framework [2].

Within this framework:

- The `CombinedLinearGaussianTransitionModel` defines a constant-velocity motion model for target prediction.

- The `LinearGaussian` or `CartesianToBearingRange` models define the linear and nonlinear measurement models, respectively.
- The `ExtendedKalmanPredictor` and `ExtendedKalmanUpdater` modules enable first-order linearization for the EKF-GM-PHD configuration.
- The `DistanceHypothesiser` and `GaussianMixtureReducer` handle measurement gating, pruning, and merging operations, using the parameters described previously.
- The `PHDUpdater` implements the recursive Bayesian update step with configurable detection and survival probabilities.

This structure allows consistent experimentation between linear and nonlinear configurations while maintaining a shared processing pipeline.

7.5.2 Preprocessing and Data Integration

For the linear KF-based GM-PHD filter, raw detections are first clustered using HDBSCAN to form point-target measurements, which are then passed to the GM-PHD filter. For the EKF-based implementation, detections are again clustered with HDBSCAN, but the resulting clusters are expressed in polar (bearing–range) coordinates before being supplied to the GM-PHD filter.

7.5.3 Performance Evaluation

Performance evaluation was conducted using the Optimal Subpattern Assignment (OSPA) metric, implemented within the Stone Soup evaluation utilities and extended for custom analysis. The OSPA metric jointly quantifies localization and cardinality errors between the estimated and ground-truth sets, providing a unified measure of multi-object tracking accuracy. In this work, OSPA was used alongside root mean square error (RMSE) metrics to compare the linear and nonlinear GM-PHD configurations under different clutter and maneuvering conditions.

7.5.4 Implementation Summary

The Stone Soup framework provided a reproducible foundation for the GM-PHD and EKF-GM-PHD filters while enabling integration with radar-specific preprocessing and evaluation tools. The modular design ensured a consistent experimental pipeline across all DLR datasets, facilitating controlled comparisons between filter configurations, parameter variations, and performance metrics.

Chapter 8

Experiments and Results

8.1 Applying the GM-PHD Filter to a Single-Target Scenario

This section establishes a baseline for single-target tracking performance with the GM-PHD filter. The single target case is one target extracted from the DLR dataset (MANV). This section also provides a reference for parameter tuning analysis.

8.1.1 Dataset Preparation and Detection Extraction

The DLR dataset does not provide explicit data association between detections and ground truth. To associate detections with the true vessel positions, the Euclidean distance between each detection and the AIS ground-truth trajectory is computed. Detections within a defined distance threshold are extracted as single-target measurements, and the corresponding AIS positions are used as the ground truth.

After applying DBSCAN to the filtered detections, the resulting cluster centroids are measurements for the GM-PHD filter, as shown in Figure 8.2. The default parameters are used initially to establish baseline behavior.

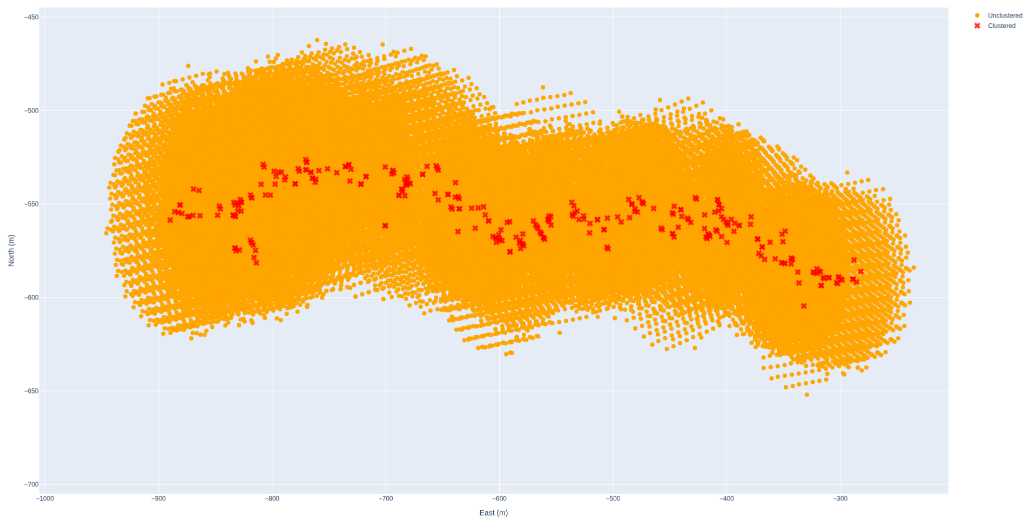

Figure 8.1: Unclustered detections (orange) and clustered detections (red) produced by the HDBSCAN algorithm.

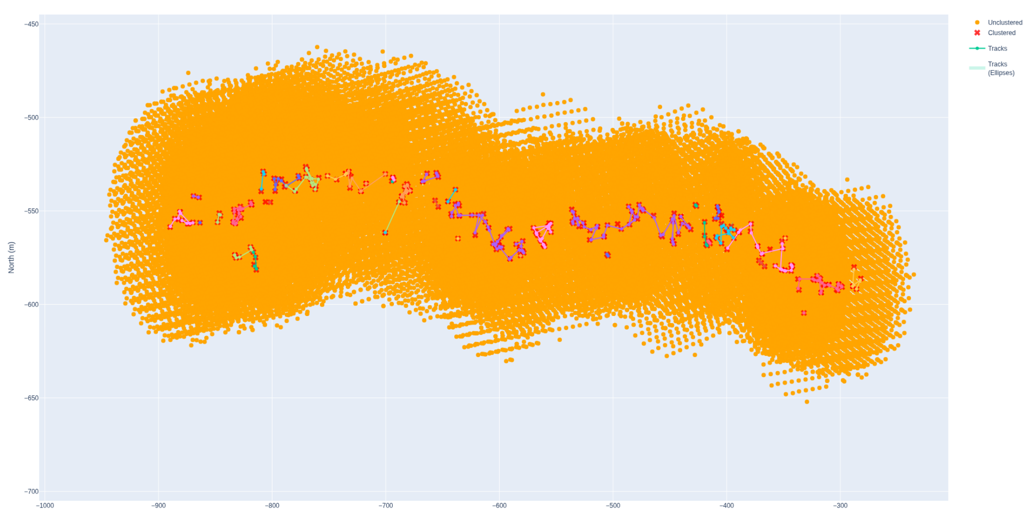

Figure 8.2: GM-PHD filter results using DBSCAN-clustered detections as input. Each color denotes a unique track.

8.1.2 Parameter Analysis

The GM-PHD filter performance depends strongly on a set of key parameters, including the merging threshold, pruning threshold, and missed distance threshold. Each parameter is tuned individually to study its impact on tracking stability and accuracy.

8.1.2.1 Merging Threshold

The merging threshold prevents an excessive number of similar Gaussian components from co-existing within the GM-PHD intensity. A low threshold may cause distinct targets to merge into a single component, while a high threshold may allow redundant components to persist, creating duplicate tracks and higher computational cost.

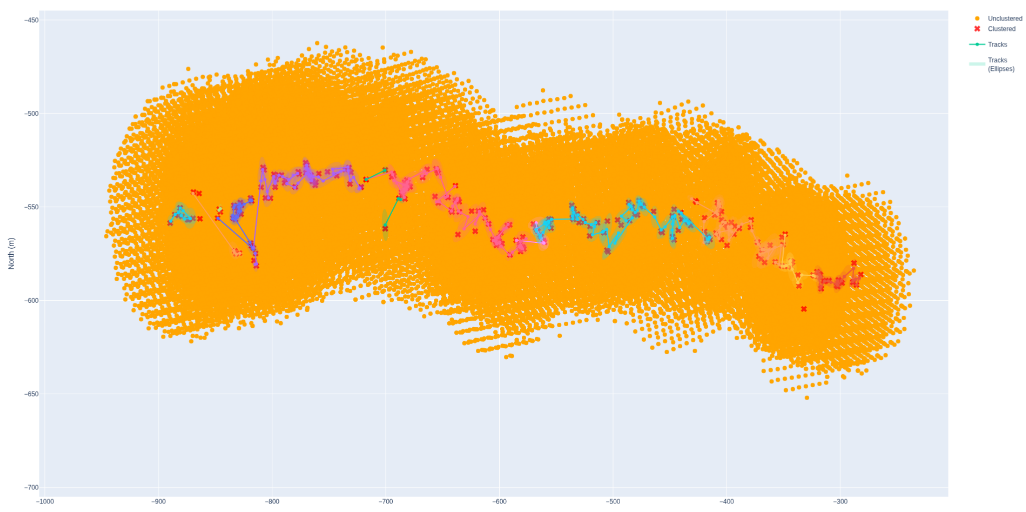

Figure 8.3: GM-PHD filter output with a merging threshold of 10 m. Overly strict merging fragments track consistency.

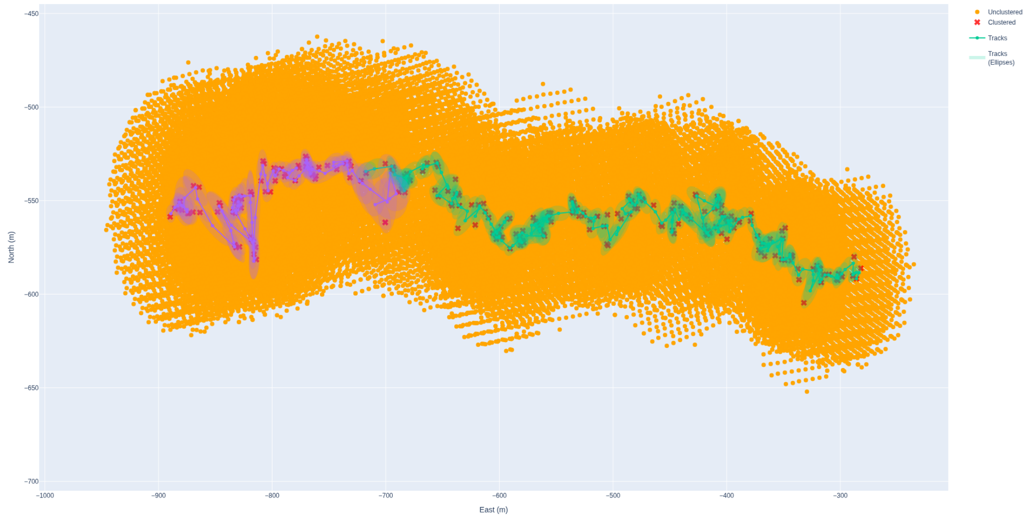

Figure 8.4: Tracks generated with a merging of 100 m. Similar Gaussian components are merged into coherent tracks, improving track consistency.

A threshold of 100 m produces more stable track estimates with fewer fragmented tracks. However, this value is dataset-dependent and may require retuning for different measurement densities.

8.1.2.2 Pruning Threshold

The pruning threshold controls the removal of Gaussian components with low weight to reduce computational complexity. If pruning is too aggressive, it may delete weak and valid tracks. If too lenient, false tracks persist.

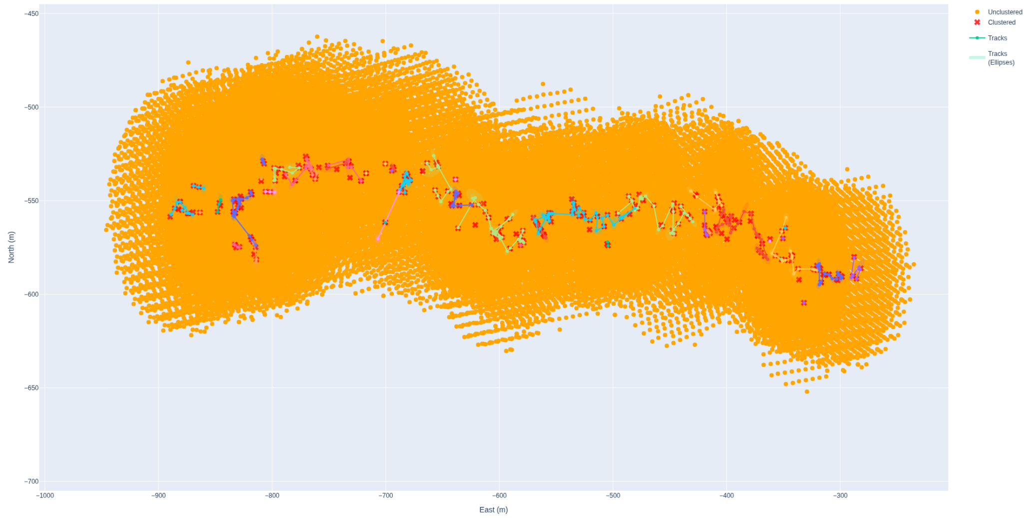

Figure 8.5: Tracks generated with a Pruning Threshold of 0.5. Excessive Track death causes fragmentation

8.1.2.3 Missed Distance Threshold

The missed-distance threshold (used during gating) defines the maximum Mahalanobis distance allowed between a predicted component and a measurement for them to be associated. If the threshold is too small, the filter may reject true detections. If it is too large, the filter may incorrectly associate detections that are far away.

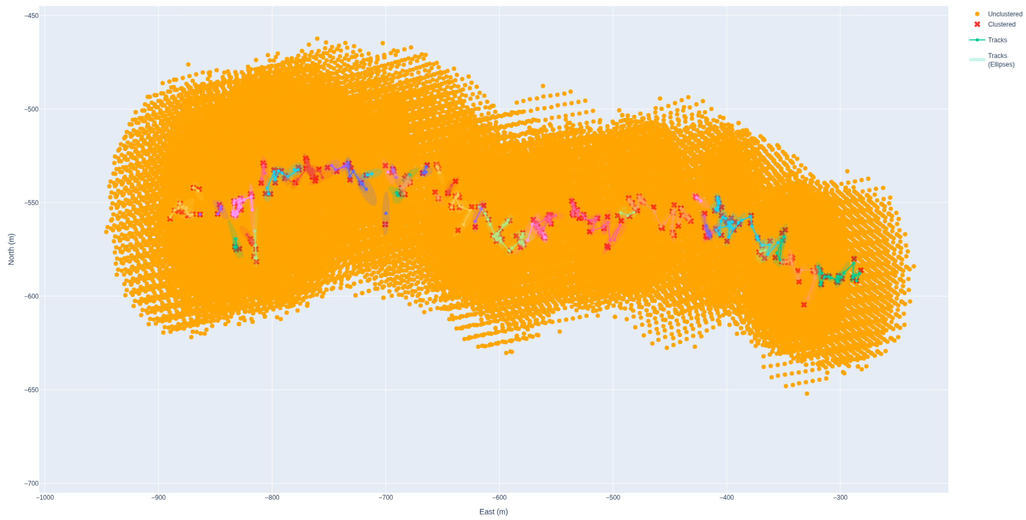

Figure 8.6: Tracks with a missed distance threshold of 3 m. Tight gating leads to excessive track generation due to frequent missed associations.

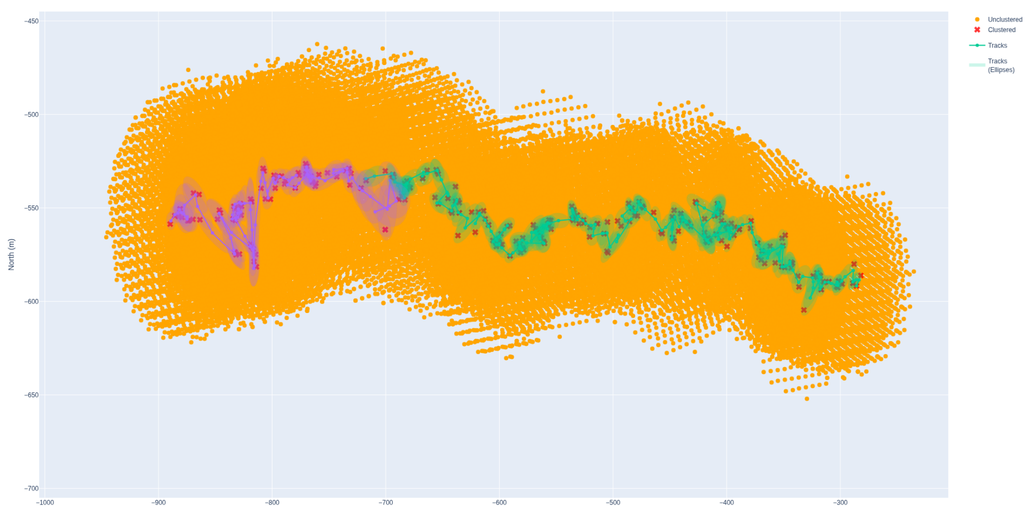

Figure 8.7: Tracks with a missed distance threshold of 50 m. Looser gating maintains more consistent tracks but increases risk of false associations.

A moderate threshold balances robustness and accuracy, maintaining coherent long-term tracks without spurious births.

8.1.3 OSPA Metrics

As detailed in Chapter 2, the OSPA metric is designed to measure the distance between sets. This is particularly important in the context of multi-object tracking (MOT), where the data association between ground truth and estimated tracks is unknown. For this test scenario, it is seen that with the optimal parameters (missed distance, merging, and pruning) the OSPA distances in meters are quite accurate with approximately 3-4 meters in error. Note that the error spike at the end is due to birth tracks at [0,0].

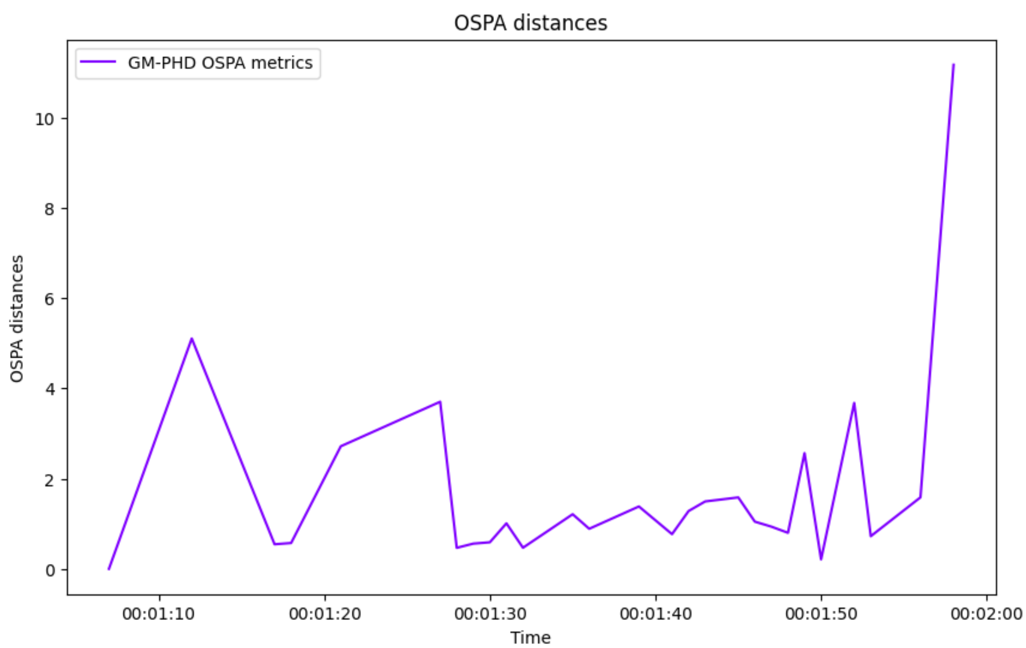

Figure 8.8: A plot of the OSPA metric applied to the GM-PHD filter results in the above scenario in meters.

8.2 Applying the GM-PHD Filter to a Multi-Object Scenario

Now that we have validated the GM-PHD filter in a single target environment. This section extends the analysis to a multi-object maritime tracking scenario using the

DLR datasets. This experiment evaluates the filter's ability to manage multiple targets, clutter, and consistent track identities over time.

8.2.1 Dataset Description and Clustering Preprocessing

The DLR dataset contains extended-target radar measurements. To satisfy the point-target assumption of the GM-PHD filter, detections originating from the same object are clustered. In the Stone Soup framework, preprocessing is often handled using k-means or DBSCAN clustering. However, these methods rely on fixed parameters that are difficult to tune with datasets with varying densities.

To address this, HDBSCAN is used instead. It adapts to regions with different densities and can form clusters without requiring the number of clusters to be assumed.

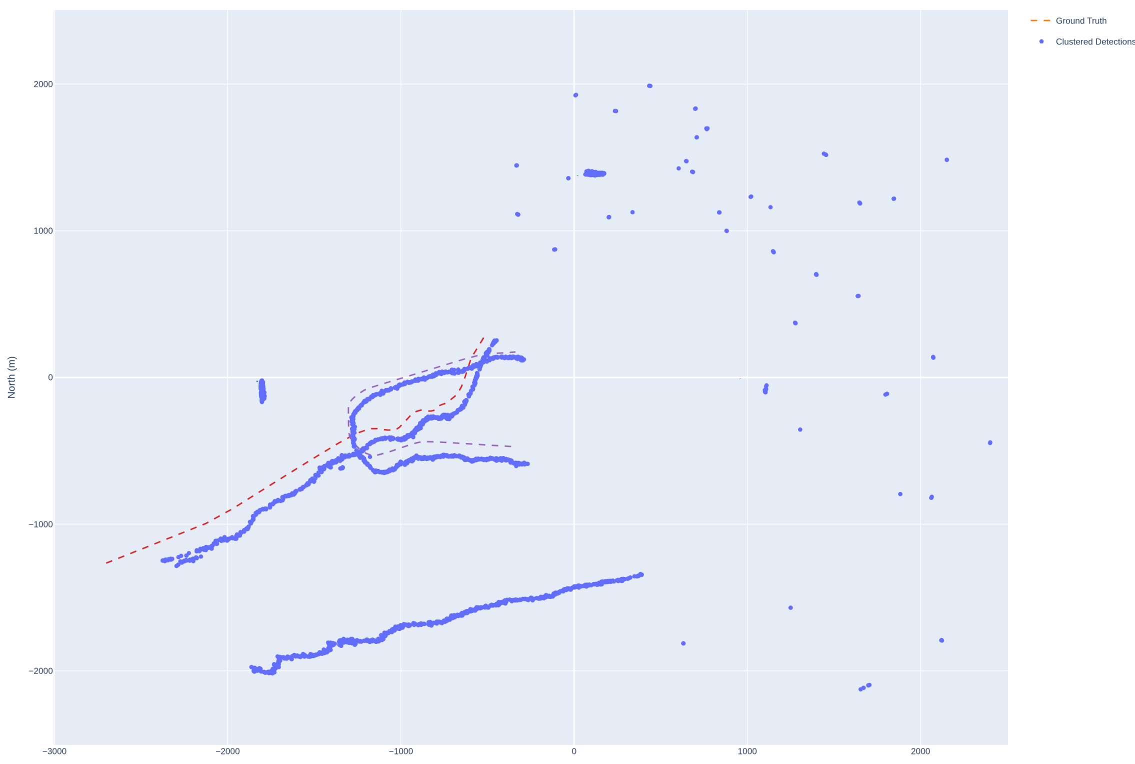

Figure 8.9: HDBSCAN-extracted clusters from the MANV dataset, showing three stationary and two moving platforms.

8.2.2 Track Generation using the GM-PHD Filter

After clustering, the resulting cluster centroids are provided as input measurements to the GM-PHD filter. Figures 8.10 and 8.11 illustrate the resulting tracks over time, where each color corresponds to a unique estimated target.

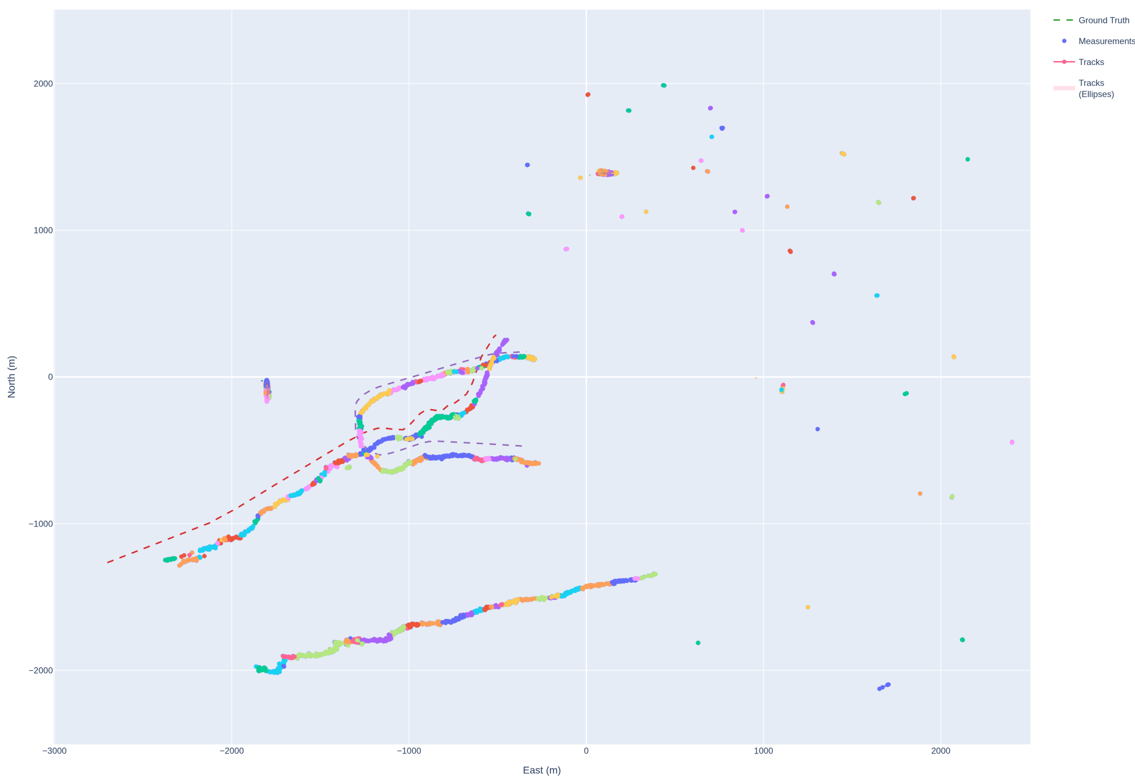

Figure 8.10: GM-PHD filter applied to DBSCAN-clustered detections. Each color represents a unique track. The visualization highlights frequent track births and pruning events, indicating over-reliance on raw detections.

The trajectories in Figure 8.10 show that the tracker frequently re-initializes tracks due to the high probability of detection parameter. When the filter assumes nearly all detections correspond to valid targets, it tends to associate new measurements too aggressively, leading to excessive track switching and fragmentation.

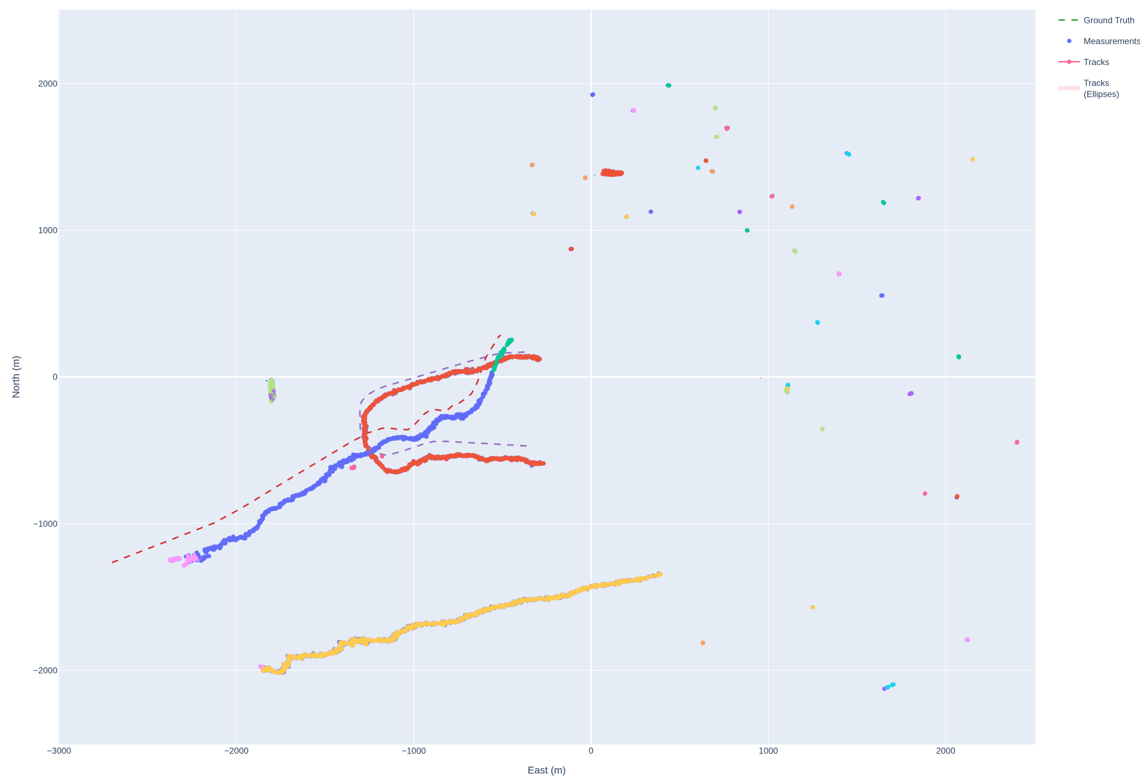

Figure 8.11: GM-PHD filter results after tuning the probability of detection parameter. Reduced track switching and smoother continuity improve overall OSPA performance.

By tuning the probability of detection to a moderate value, the GM-PHD filter relies more heavily on motion prediction during periods of sparse detections. This adjustment mitigates false track initiations from clutter and stabilizes long-term trajectories, directly improving the OSPA metric.

8.2.3 Quantitative Evaluation using RMSE Metrics

To quantitatively assess the tracking performance, the root mean square error (RMSE) is first examined. In Figure 8.12, adapted from [1], Targets 2 and 3 exhibit relatively accurate positional estimates. However, it should be noted that the Automatic Identification System (AIS) data, used as the ground truth in this dataset, is not perfectly accurate, as discussed in the referenced study. Consequently, position errors in the range of 100–200 meters are expected due to inherent AIS offsets, as previously illustrated.

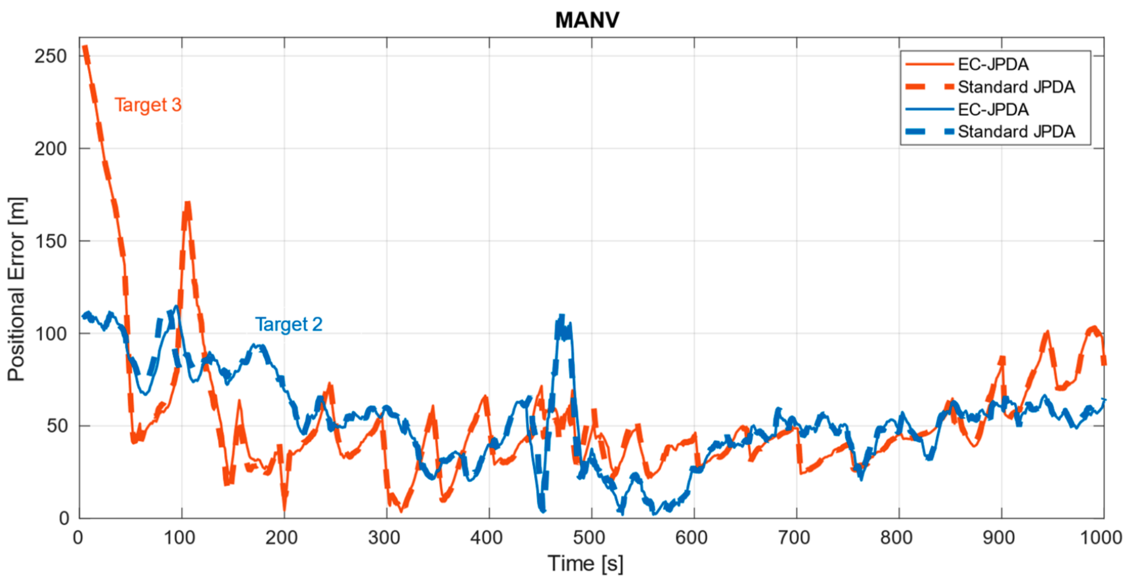

Figure 8.12: RMSE position error (m) for Targets 2 and 3, adapted from [1].

Figure 8.13: RMSE position error (m) for the PHD Tracker implementation.

When compared to the PHD Tracker results in Figure 8.13, the overall range and behavior of the RMSE values appear similar, again reflecting the influence of AIS inaccuracies. The spikes in RMSE reaching approximately 250 meters are likely caused by track terminations and subsequent regenerations, which temporarily degrade positional continuity at around Time = 900 seconds. Overall, the results indicate that the PHD Tracker achieves comparable tracking performance to the JPDA Tracker, despite operating under less restrictive assumptions. Specifically, the PHD Tracker does not rely on prior knowledge of target birth locations or the number of active targets at any given time. In contrast, the JPDA Tracker assumes a fixed number of targets and utilizes k-means clustering initialized with the known number of clusters to associate measurements with tracks. This demonstrates the PHD Tracker's ability to effectively estimate target states and maintain robust performance even in scenarios with uncertain or dynamic target counts.

8.3 Applying the GM-PHD Filter to a Cluttered Scenario

Having demonstrated the GM-PHD filter's multi-target tracking capability in the controlled conditions using the MANV dataset, this section evaluates its robustness in the presence of clutter and interference. For this purpose, the DARC (Data Association with RACON) dataset from the DLR measurement campaigns is used.

8.3.1 Impact of Clutter on Clustering Performance

The RACON reflections have a pronounced impact on the performance of density-based clustering algorithms such as DBSCAN and HDBSCAN. Due to the RACON reflections being a similar density to the target detection density, DBSCAN with its fixed *eps* parameter, tends to cluster these detections that do not correspond to real targets.

HDBSCAN demonstrates improved robustness compared to other methods, as its hierarchical structure allows it to adapt to varying density levels within the dataset. However, still struggles with clutter density similar to that of the target ship.

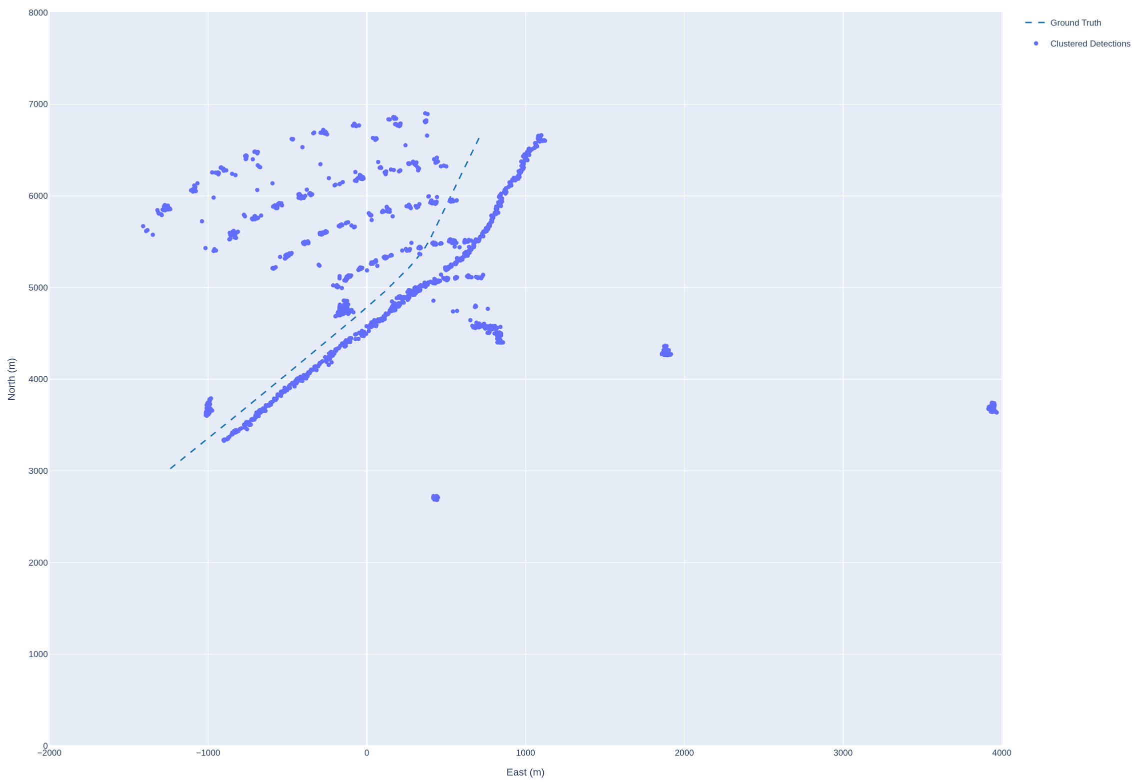

Figure 8.14: HDBSCAN clustering results on the DARC dataset with plotted ground truth.

In Figure 8.14, the vessel of interest begins its trajectory at approximately $(x, y) = (1000\ \text{m},\ 6500\ \text{m})$ and proceeds southward along a stable path. Surrounding this trajectory, a cone-shaped region of dense detections is visible, spanning approximately

$$x \in [-1500,\ 500]\ \text{m}, \quad y \in [4500,\ 7000]\ \text{m}.$$

This area corresponds to RACON-induced clutter, which appears as strong and persistent radar reflections that form a dense cluster of points. Because this clutter occupies the same general region as the vessel's trajectory, HDBSCAN may assign these points to clusters that resemble valid detections. Since this clutter is still clustered, final clutter rejection must be handled within the GM-PHD filtering stage.

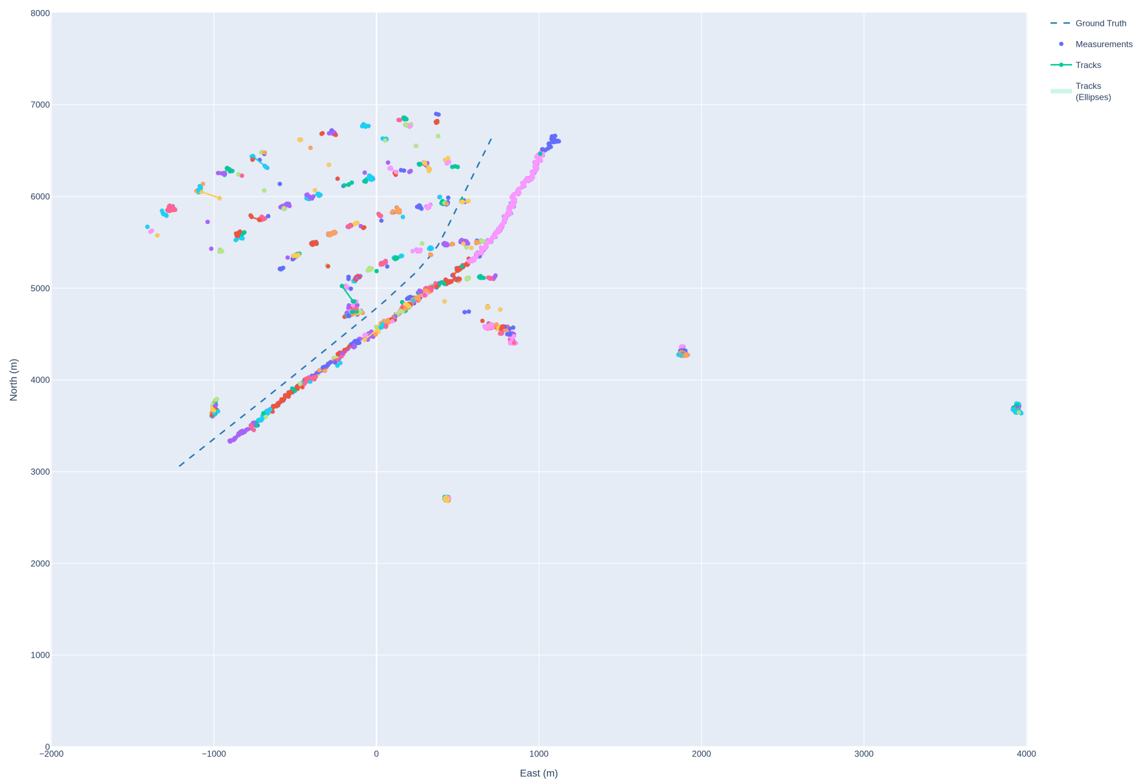

Figure 8.15: GM-PHD filter results on the DARC dataset, evaluated without modeling clutter (clutter spatial density set to zero). The plotted ground truth trajectories are shown for reference.

In Figure 8.15, it can be observed that the absence of a clutter model causes the GM-PHD filter to generate a significant number of spurious tracks, particularly within the RACON-affected region. The high density of false detections in this area leads to the formation of short-lived tracks that do not correspond to real targets. Moreover, the clutter interferes with the correct propagation of the true target's track, resulting in loss of continuity along its trajectory. This behavior highlights the importance of explicitly modeling clutter spatial density within the GM-PHD framework to suppress false track initiations and improve the stability of target tracking in environments with high radar interference.

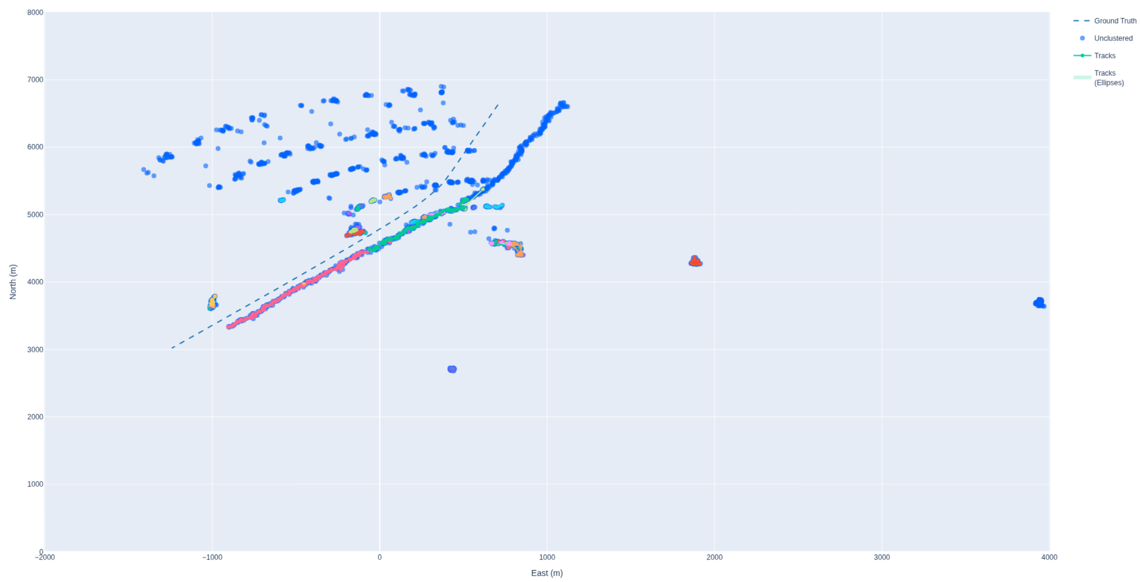

Figure 8.16: GM-PHD filter results on the DARC dataset, evaluated with modeling clutter. The plotted ground truth trajectories are shown for reference.

In Figure 8.16, the clutter suppression performance is improved, but the filter fails to initiate a track at the start of the trajectory. This is likely because the early target detections have a spatial density similar to that of the clutter. This behavior illustrates the trade-off in selecting the clutter spatial density parameter for a given dataset.

8.3.2 Application of Nonlinear Tracking Filter

In cluttered maritime radar environments such as those in the DARC dataset, using the native bearing–range measurements can help preserve the radar's sensing geometry. To accommodate these nonlinear measurements, the GM-PHD filter is implemented with an Extended Kalman Filter (EKF) update.

The EKF linearizes the nonlinear bearing–range measurement function around the predicted state, enabling the tracker to correctly incorporate angular and range uncertainties without relying on a Cartesian approximation. This should theoretically improve the tracking performance due to a more accurate covariance model.

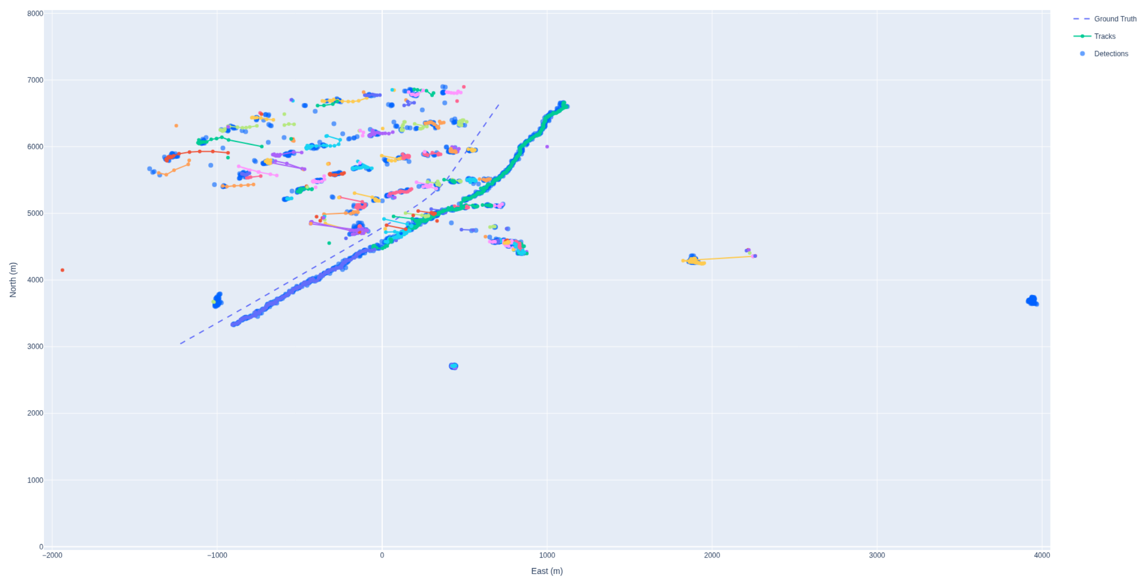

Figure 8.17: Extended Kalman GM-PHD filter results on the DARC dataset with non-zero clutter spatial density. The plotted ground truth trajectories are shown for reference.

In Figure 8.17 it can be seen that there is stable track consistently along the target's path, but the filter fails to suppress the clutter in the region specified earlier. Note that to achieve this performance, it was required to have the birth component be fairly close to the region of detections:

$$\mathbf{m}_0 = \begin{bmatrix} 6000 \\ 10 \\ 1000 \\ 10 \end{bmatrix}, \qquad \mathbf{P}_0 = \begin{bmatrix} 10^6 & 0 & 0 & 0 \\ 0 & 400 & 0 & 0 \\ 0 & 0 & 10^6 & 0 \\ 0 & 0 & 0 & 400 \end{bmatrix}.$$

where $\mathbf{m}_0$ is the initial state vector and $\mathbf{P}_0$ is the corresponding initial covariance matrix. Suboptimal performance may result from inadequate convergence when the EKF is linearized about a prior that is significantly displaced from the true state, causing the filter to diverge or converge slowly.

8.3.3 Error Metrics

It can be observed in Figure 8.18 that the JPDA Tracker exhibits RMSE values fluctuating between approximately 200 and 300 meters. In comparison, the PHD Tracker results shown in Figure 8.19 reveal that the first stable track is initiated at around $t = 100$ seconds. After track initialization, the PHD Tracker maintains an average positional error of approximately 400 meters, with occasional spikes corresponding to instances of spurious track generation or temporary misassociations near the ground truth trajectory.

The slightly higher average RMSE observed in the PHD Tracker can be attributed to several factors. First, the characteristics of the clustering algorithm used during measurement association may introduce uncertainty in target grouping, particularly in dense or noisy measurement environments. Second, the inherent inaccuracies in the (AIS) data used as the ground truth introduce additional uncertainty when assessing true positional accuracy. As noted in the reference, AIS-reported positions can deviate due to detections over the target's surface and antenna placement [1].

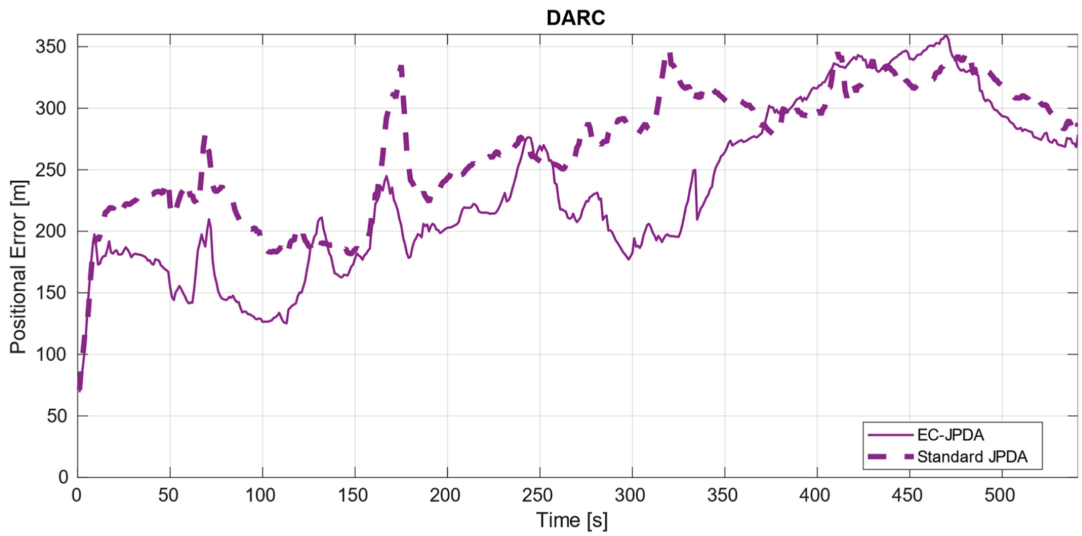

Figure 8.18: Comparison of root mean square error (RMSE) performance on the DARC maritime dataset using the JPDA filter.

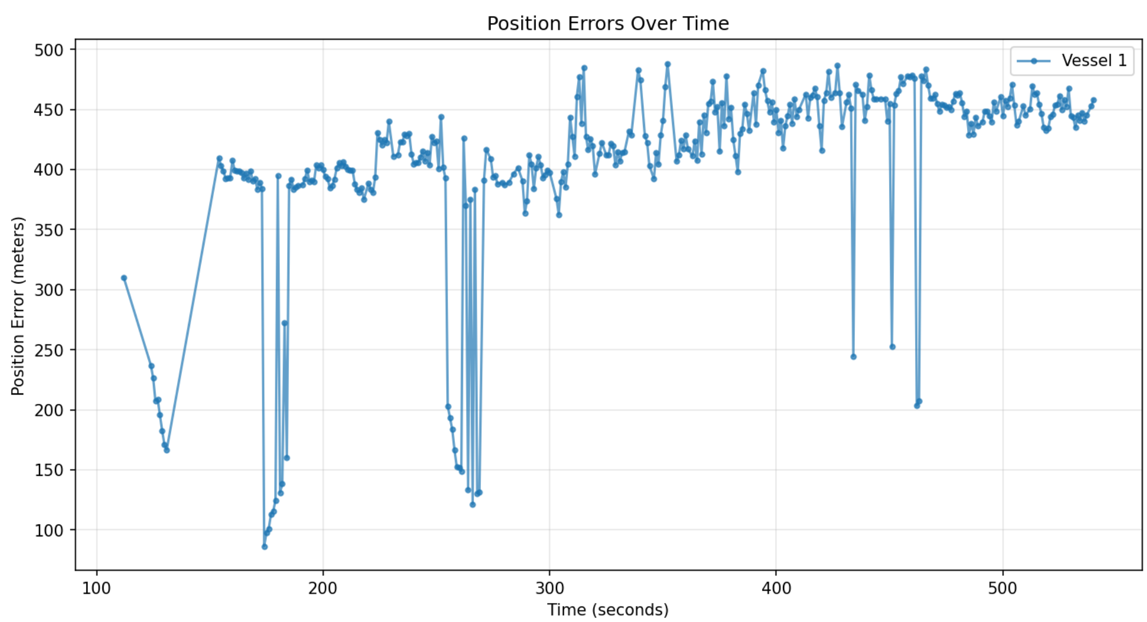

Figure 8.19: RMSE performance on the DARC maritime dataset using the GM-PHD filter implementation.

Given these limitations, it is reasonable to interpret the PHD Tracker's performance as consistent with the underlying detection data rather than the nominal AIS ground truth. Therefore, accuracy relative to radar detections may serve as a more meaningful indicator of tracking performance in this dataset, since the radar's range resolution and measurement precision are independent of the tracker's ability to estimate target motion and maintain track continuity.

Chapter 9

PHD-SLAM (Simultaneous Localization and Mapping)

9.1 Problem Formulation of SLAM with Random Finite Sets

Having established the fundamental principles of SLAM, it is now necessary to examine its probabilistic structure and to introduce the Random Finite Set (RFS) formulation that serves as the foundation of the PHD-SLAM framework. This section presents the evolution from classical Bayesian SLAM to the random-set representation, demonstrating how the latter naturally overcomes key limitations of the former. Note that this section's theory is based on the formulations of two papers [6][29].

9.1.1 Probabilistic Model of SLAM

The SLAM problem can be expressed as the joint estimation of the robot trajectory and the map of landmarks, conditioned on all sensor observations and control inputs up to a given time step. Following the Bayesian perspective described in [30], the objective is to determine the posterior distribution

$$P(\mathbf{X}_{0:t}, \mathbf{m} \mid \mathbf{Z}_{1:t}, \mathbf{U}_{1:t}, \mathbf{X_0}) \tag{9.1}$$

where $\mathbf{X}_{0:t} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t\}$ denotes the sequence of robot poses up to time t, $\mathbf{m} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_n\}$ represents the set of landmark positions, $\mathbf{U}_{1:t} = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_t\}$ contains the control inputs or odometry readings, and $\mathbf{Z}_{1:t} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_t\}$ represents the set of landmark observations collected by the robot sensors. The estimation of this posterior is typically performed recursively as new motion and observation data become available. The probabilistic formulation of SLAM relies on two fundamental components. The first is the motion model,

$$P(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_t), \tag{9.2}$$

which represents the probability of transitioning from state $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$ under the applied control $\mathbf{u}_t$. The second is the observation model,

$$P(\mathbf{z}_t \mid \mathbf{x}_t, \mathbf{m}), \tag{9.3}$$

which expresses the likelihood of obtaining a particular sensor measurement $\mathbf{z}_t$ given the robot pose $\mathbf{x}_t$ and the positions of landmarks $\mathbf{m}$. By applying Bayes' rule to Eq. (9.1) and invoking the Markov assumption that the current state depends only on the previous state, the recursive expression for the SLAM posterior becomes

$$\begin{aligned} P(\mathbf{X}_{0:t}, \mathbf{m} \mid \mathbf{Z}_{1:t}, \mathbf{U}_{1:t}) &= \alpha \, P(\mathbf{z}_t \mid \mathbf{x}_t, \mathbf{m}) \, P(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_t) \\ &\quad \times P(\mathbf{X}_{0:t-1}, \mathbf{m} \mid \mathbf{Z}_{1:t-1}, \mathbf{U}_{1:t-1}) \end{aligned} \tag{9.4}$$

where α is a normalizing constant [30]. Equation (9.4) provides the basis for many modern SLAM algorithms, including the Extended Kalman Filter SLAM and Fast-SLAM, both of which iteratively apply this recursive structure to maintain a joint estimate of robot pose and map uncertainty.

9.1.2 Limitations of Classical SLAM

Although probabilistic SLAM algorithms have been successfully applied across numerous domains, they rely on several simplifying assumptions that become restrictive in realistic sensing environments. Conventional SLAM assumes that the number of landmarks is known and fixed, that each observation can be explicitly associated with a unique landmark through data association, and that each landmark generates at most one measurement per time step. These assumptions hold for small-scale or structured environments but degrade in performance in the presence of dense clutter, uncertain detection, or extended targets that produce multiple measurements simultaneously.

In such cases, new landmarks may appear or disappear over time, spurious detections may arise from clutter or noise, and the association between measurements and landmarks becomes ambiguous. Under these conditions, the conventional vector representation of the map and observations imposes both computational and conceptual limitations. It requires the filter to track each landmark explicitly and maintain a fixed vector ordering, even when the true number of features varies dynamically. To address these challenges, the SLAM formulation can be extended into the Random Finite Set (RFS) framework, which models both the map and the set of measurements as random finite sets whose cardinalities and elements evolve stochastically.

9.1.3 Random Finite Set Representation

In classical formulations, landmarks and measurements are represented as ordered vectors whose element indices correspond to specific physical features. This ordering creates a dependency on the robot's viewpoint and requires a data association step to ensure that each measurement $\mathbf{z}_i$ is matched to its corresponding landmark $\mathbf{m}_j$. The RFS formulation removes this dependency by representing both quantities as finite-valued random sets. At time t, the map and measurement spaces are expressed as

$$\mathcal{M}_t = \{m_1, m_2, \ldots, m_{N_t}\}, \qquad \mathcal{Z}_t = \{z_1, z_2, \ldots, z_{M_t}\}, \tag{9.5}$$

where N_t and M_t are random variables representing the number of landmarks and measurements, respectively. In this representation, the ordering of elements within $\mathcal{M}_t$ and $\mathcal{Z}_t$ is immaterial. For example, a measurement vector defined as

$$Z = [z_1, z_2, z_3, z_4, z_5]^\top \quad \text{or} \quad Z = [z_5, z_4, z_3, z_2, z_1]^\top \tag{9.6}$$

corresponds to the same random finite set $\mathcal{Z} = \{z_1, z_2, z_3, z_4, z_5\}$. This unordered property eliminates the need for explicit data association and enables a natural mechanism for handling the appearance and disappearance of landmarks. The cardinality of the set may increase or decrease over time, and operations such as union and intersection can be used to represent feature births, deaths, and merged detections in a mathematically consistent manner.

9.1.4 Theoretical Framework of PHD-SLAM

The original Probability Hypothesis Density (PHD)–SLAM algorithm extends the Random Finite Set (RFS) formulation to the simultaneous localization and mapping problem by jointly estimating the robot pose and the intensity function of the surrounding [6]. Conceptually, PHD-SLAM combines a particle filter for robot localization with a Gaussian–mixture PHD filter for mapping, thereby performing multi-target estimation without explicit data association. The algorithm can be interpreted as a Rao-Blackwellised particle filter in which the nonlinear vehicle motion is represented by samples, while the map, conditioned on each pose hypothesis, is estimated analytically through the propagation of Gaussian mixture components.

Formally, the state of the system at time t is represented by the pair $(\mathbf{x}_t, \mathcal{M}_t)$, where $\mathbf{x}_t$ is the robot pose and $\mathcal{M}_t = \{m_1, \dots, m_{N_t}\}$ is the random finite set of landmarks. The joint posterior density of these two components, conditioned on all controls and measurements up to time t, is expressed as

$$P(\mathbf{x}_t, \mathcal{M}_t \mid \mathbf{Z}_{1:t}, \mathbf{U}_{1:t}) \propto P(\mathbf{Z}_t \mid \mathbf{x}_t, \mathcal{M}_t)\, P(\mathbf{x}_t, \mathcal{M}_t \mid \mathbf{Z}_{1:t-1}, \mathbf{U}_{1:t}), \tag{9.7}$$

where $\mathbf{Z}_t$ and $\mathbf{U}_t$ denote the measurement and control sets, respectively. The exact computation of (9.7) is intractable because the number of landmarks is unknown and changes over time. The PHD-SLAM approach therefore approximates this posterior by its first-order statistical moment, the *intensity* or Probability Hypothesis Density (PHD) function $v_t(x)$, which represents the expected spatial density of landmarks in the environment.

The PHD follows a recursive Bayesian filtering structure comprising a prediction and an update stage. In the prediction step, surviving landmarks from the previous time step and newly born landmarks are propagated according to

$$v_{t|t-1}(m) = p_S \int f_{t|t-1}(m \mid m')\, v_{t-1}(m')\, dm' + \gamma_t(m), \tag{9.8}$$

where p_S is the probability of landmark survival, $f_{t|t-1}(\cdot)$ is the landmark transition model, and $\gamma_t(m)$ is the birth intensity describing the rate and spatial distribution of newly detected landmarks. The robot pose for each particle i is predicted through the nonlinear motion model

$$\mathbf{x}_t^{(i)} = f(\mathbf{x}_{t-1}^{(i)}, \mathbf{u}_t) + \mathbf{q}_t, \tag{9.9}$$

where $\mathbf{q}_t$ is zero-mean Gaussian motion noise. During the update step, the PHD intensity is corrected using the current measurement set $\mathbf{Z}_t$. Given the detection probability P_D and the clutter intensity $\kappa(z)$, the updated intensity is given by

$$v_t(m) = (1 - P_D)\, v_{t|t-1}(m) + \sum_{z \in \mathbf{Z}_t} \frac{P_D\, g(z \mid m)\, v_{t|t-1}(m)}{\kappa(z) + \int P_D\, g(z \mid m')\, v_{t|t-1}(m')\, dm'}, \quad (9.10)$$

where $g(z \mid m)$ denotes the measurement likelihood function. Each term in the summation corresponds to the contribution of a measurement z to the posterior intensity. In practice, the intensity $v_t(m)$ is represented as a Gaussian mixture of the form

$$v_t(m) = \sum_{j=1}^{J_t} w_t^{(j)}\, \mathcal{N}(m; m_t^{(j)}, P_t^{(j)}), \quad (9.11)$$

where each component weight $w_t^{(j)}$ represents the existence probability of a potential landmark, and $(m_t^{(j)}, P_t^{(j)})$ denote its estimated mean and covariance. Pruning and merging operations are applied after each update to limit the number of components and maintain computational tractability.

At the higher level, the robot pose distribution is maintained using a particle filter. Each particle carries its own local map intensity, which is updated according to Eqs. (9.8) and (9.10). The importance weight of particle i is proportional to the likelihood of the current measurement set under its associated map,

$$\omega_t^{(i)} \propto \omega_{t-1}^{(i)}\, P(\mathbf{Z}_t \mid \mathbf{x}_t^{(i)}, v_t^{(i)}), \quad (9.12)$$

and the particles are resampled periodically to focus computational effort on the most probable trajectories. The overall joint estimate of robot pose and map can then be obtained by taking the weighted mean of the particle states and the mixture components across the ensemble.

This formulation preserves the strengths of both particle filtering and random finite set statistics. The particle filter captures the nonlinear and non-Gaussian characteristics of vehicle motion, while the Gaussian-mixture PHD analytically represents the uncertainty and cardinality of map features without requiring explicit data association. Consequently, PHD-SLAM provides a principled framework for performing SLAM in the presence of false alarms, missed detections, and an unknown or time-varying number of landmarks.

9.1.5 Theoretical Framework of PHD-SLAM 2.0

Building upon the original PHD-SLAM formulation, Gao *et al.* [29] proposed an improved version, referred to as PHD-SLAM 2.0, to address the inefficiency of the standard algorithm in highly uncertain or cluttered environments. The central innovation of PHD-SLAM 2.0 lies in the design of a new proposal distribution (PD) for vehicle pose sampling that explicitly incorporates the most recent sensor measurements, thereby reducing the number of particles required for accurate inference and significantly improving computational efficiency.

In standard PHD-SLAM, the robot pose distribution is propagated using a particle filter whose proposal distribution depends solely on the previous pose and control input, i.e.,

$$q_t(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) = \phi_t(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_{t-1}), \tag{9.13}$$

where $\phi_t(\cdot)$ denotes the Markov transition density of the motion model. While this simplifies implementation, it ignores the information contained in the current measurements, making the filter highly dependent on the motion prior. Consequently, a vast number of particles are typically required to achieve sufficient coverage of the pose space, especially when odometry is noisy or when the vehicle undergoes abrupt motion.

PHD-SLAM 2.0 introduces a measurement-driven proposal that approximates the marginal posterior distribution of the vehicle pose using the joint PHD of the vehicle and map states. Denoting by $D_t(\mathbf{x}, \mathbf{m})$ the joint PHD of vehicle pose $\mathbf{x}$ and landmark $\mathbf{m}$ conditioned on all measurements up to time t, the marginal pose density is obtained by integration over the map state:

$$p_t(\mathbf{x}) \approx \frac{\int D_t(\mathbf{x}, \mathbf{m})\, d\mathbf{m}}{\iint D_t(\mathbf{x}, \mathbf{m})\, d\mathbf{x}\, d\mathbf{m}}. \tag{9.14}$$

This marginal is expressed in Gaussian mixture form, where each component encodes a local hypothesis of the vehicle state consistent with one of the most recent measurements. For each particle, the mean and covariance of this Gaussian mixture are updated through Kalman filter–like operation that couples the predicted pose uncertainty with the measurement likelihood function. The resulting Gaussian density

$$q_t^{(i)}(\mathbf{x}_t | \mathbf{x}_{t-1}^{(i)}, \mathbf{u}_{t-1}, \mathbf{Z}_t) = \mathcal{N}(\mathbf{x}_t;\, \hat{\mathbf{x}}_{t|t}^{(i)},\, \Sigma_{t|t}^{(i)}) \tag{9.15}$$

is then adopted as the new proposal distribution for particle i. Sampling from (9.15) rather than from the motion prior ensures that particles are concentrated in regions supported by both motion and measurement information.

The importance weight associated with each particle is adjusted to account for the new proposal, according to

$$\omega_t^{(i)} \propto \omega_{t-1}^{(i)} \frac{\phi_t(\hat{\mathbf{x}}_t^{(i)} | \hat{\mathbf{x}}_{t-1}^{(i)}, \mathbf{u}_{t-1})}{q_t^{(i)}(\hat{\mathbf{x}}_t^{(i)} | \hat{\mathbf{x}}_{t-1}^{(i)}, \mathbf{u}_{t-1}, \mathbf{Z}_t)}. \tag{9.16}$$

This update compensates for the difference between the motion prior and the new measurement-driven proposal. The conditional map densities, represented by Gaussian Mixture PHDs, are then updated analytically as in standard PHD-SLAM through the standard prediction and measurement-update steps. Each particle therefore carries its own conditional PHD map and updated pose estimate.

By coupling pose and measurement information in the proposal, PHD-SLAM 2.0 achieves the same estimation accuracy as standard PHD-SLAM with far fewer particles. The algorithm retains linear computational complexity with respect to the number of features and measurements within the field of view, while the per-particle cost increases only marginally due to the covariance propagation of the proposal. In practice, this yields an overall reduction in computation time and improved real-time feasibility, especially in cluttered environments or under low detection probabilities.

Theoretically, PHD-SLAM 2.0 can be interpreted as a Rao–Blackwellised particle filter in which each particle maintains a Gaussian-mixture map updated by a PHD filter, and the vehicle pose is sampled from a posterior approximation that fuses motion and measurement information. This integration allows the algorithm to remain robust to missed detections, false alarms, and measurement origin uncertainty, while drastically reducing the number of required particles compared to the original PHD-SLAM formulation.

9.2 Application of PHD-SLAM 2.0

To evaluate the performance of the proposed radar-based SLAM approach, the PHD-SLAM 2.0 framework was employed to process synthetic radar data generated in the Stone Soup environment. While the DLR maritime datasets served as the primary benchmark for the tracking experiments in previous chapters, these scenarios consist predominantly of dynamic targets. Since standard SLAM formulations rely on stationary landmarks to constrain the vehicle's pose estimate, the lack of static features in the open-ocean DLR data renders it unsuitable for this specific evaluation. Consequently, a controlled synthetic environment was utilized to demonstrate theoretical feasibility and to validate the ability of a Probability Hypothesis Density–based SLAM algorithm to jointly estimate both the vehicle trajectory and the spatial distribution of landmarks under realistic radar noise conditions.

9.2.1 Synthetic Dataset Generation in Stone Soup

A synthetic maritime radar dataset was generated using a moving sensor platform following a perfect circular trajectory of radius 400 m about the origin with a constant turn rate of $\omega = 0.05$ rad/s. This produces a vessel path with known pose and velocity at each simulation step. A static bearing–range radar was attached to the platform with a field of view of 90° and a maximum range of 2000 m. The measurement noise covariance was set to

$$R = \begin{bmatrix} \sigma_\theta^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix} = \begin{bmatrix} (1^\circ)^2 & 0 \\ 0 & (10\ \mathrm{m})^2 \end{bmatrix},$$

corresponding to an azimuth standard deviation of 1° and a range standard deviation of 10 m. This choice approximates a moderately noisy radar; in practice, the true covariance of a radar system can vary depending on hardware and environment.

The environment was populated with stationary targets that served as landmarks. Several landmarks were placed deterministically at ranges between 500 m and 800 m in different quadrants, and additional landmark positions were drawn uniformly at random from a square region $[-1000, 1000] \times [-1000, 1000]$ m to create a more heterogeneous and cluttered scene. Each landmark was implemented as a zero-velocity constant-velocity model, yielding effectively static point targets in the global Cartesian frame.

At each simulation step, the radar generated noisy bearing–range detections of all landmarks within its field of view. False alarms were modeled using a Poisson clutter process with an average rate of 1.5 clutter detections per dwell over the region:

$$x, y \in [-2000, 2000] \text{ m},$$

resulting in a realistic mixture of target-originated detections and spurious returns. The simulation produced the following dataset components:

1. **Sensor platform ground truth**: time-stamped vessel states in Cartesian coordinates (x, y) with velocities (v_x, v_y) and turn rate ω

2. **Landmark ground truth**: stationary landmarks

3. **Radar detections**: time-stamped bearing–range measurements with Gaussian noise R, including both target detections and labeled clutter

4. **Odometry signals**: true speed and turn rate (v_k, ω_k) and *noisy odometry* generated by adding independent zero-mean Gaussian noise,

$$\tilde{v}_k = v_k + n_{v,k}, \quad n_{v,k} \sim \mathcal{N}(0, \sigma_v^2), \quad \sigma_v = 0.64 \text{ m/s},$$

$$\tilde{\omega}_k = \omega_k + n_{\omega,k}, \quad n_{\omega,k} \sim \mathcal{N}(0, \sigma_\omega^2), \quad \sigma_\omega = 0.02 \text{ rad/s}.$$

   These signals emulate noisy speed and heading-rate measurements used as odometry inputs.

The CSV files were subsequently combined into a MATLAB `.mat` file for use in the PHD-SLAM 2.0 processing pipeline.

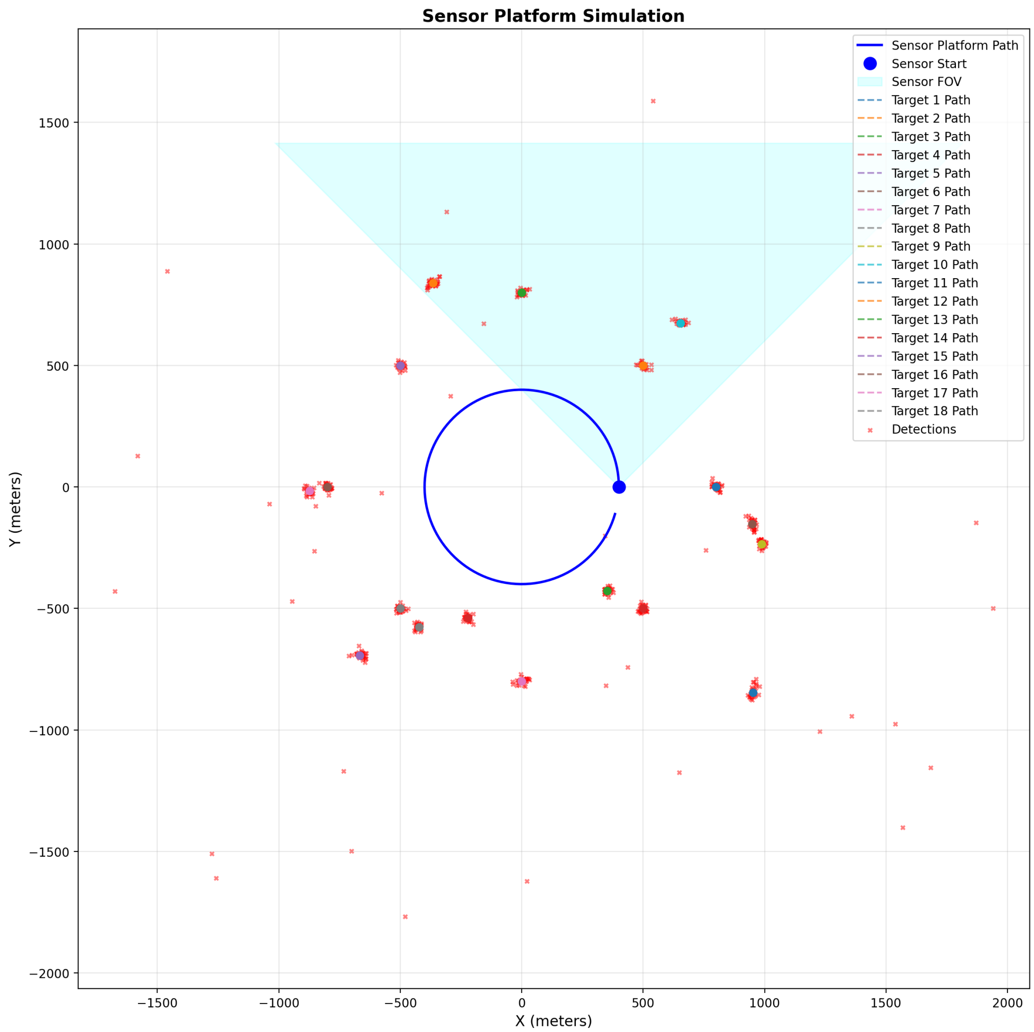

Figure 9.1: Simulated circular radar platform with stationary landmarks, detections, and a 90° field of view.

9.2.2 Integration with PHD-SLAM 2.0

The MATLAB implementation of PHD-SLAM 2.0 was used to process the Stone Soup-generated detections. This framework implements a Random Finite Set (RFS) based formulation of Simultaneous Localization and Mapping, in which the map is represented as an intensity function over landmark states rather than a fixed list of feature points. The algorithm performs three primary operations at each time step:

1. Prediction: Propagate the joint vehicle and landmark intensity using a motion model and process noise covariance

2. Update: Incorporate radar measurements to refine the posterior intensity, updating both vehicle pose and landmark distributions

3. Resampling and pruning: Maintain tractable computational complexity by pruning low weight hypotheses and merging components

Ground-truth vehicle poses and detections exported from Stone Soup were used to initialize the filter and to evaluate the consistency of the estimated trajectory.

9.2.3 Evaluation and Visualization

The resulting output was compared against the ground-truth trajectory and landmark positions from the synthetic dataset. The reconstructed map demonstrates that the algorithm can accurately recover landmark locations and maintain stable pose estimates even under moderate measurement noise. Figure 9.2 shows the estimated vehicle path together with the reconstructed landmark map. Overall, the estimated map aligns closely with the ground-truth structure, indicating that the probabilistic framework effectively integrates radar detections into consistent spatial estimates.

Minor discrepancies between the estimated and true landmark positions are observed in the later time steps. This deviation likely arises from accumulated pose uncertainty in the estimate.

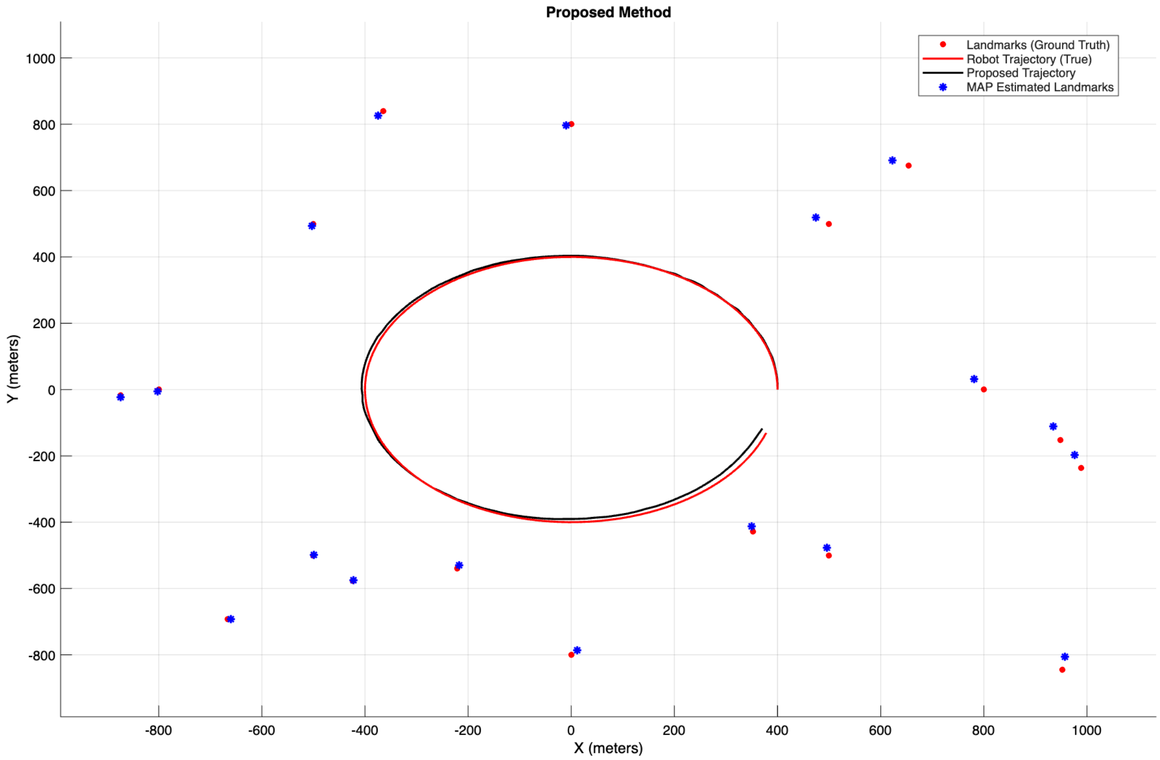

Figure 9.2: Red circles denote ground-truth landmarks, blue stars denote MAP-estimated landmark positions, the red line indicates the true platform trajectory, and the black line shows the PHD-SLAM 2.0 trajectory estimate for the synthetic dataset.

The evolution of trajectory accuracy over time is evaluated using the root-mean-square error (RMSE). Figure 9.3 shows the position RMSE over time for both a dead-reckoning baseline and the PHD-SLAM 2.0 estimate. The dead-reckoning trajectory drifts over time, whereas the PHD-SLAM 2.0 estimate remains bounded and exhibits smaller error once sufficient radar measurements are available. This shows that adding radar measurements improves the state estimate, as theorized in PHD-SLAM 2.0, compared to using odometry alone.

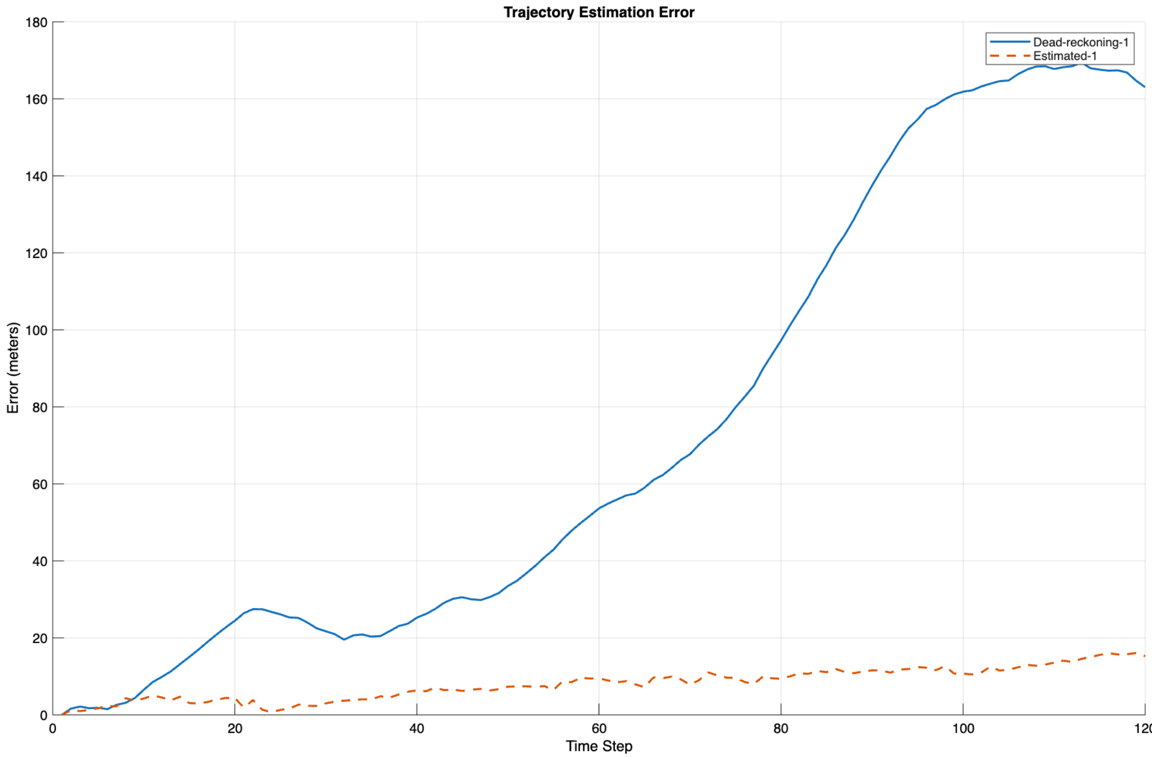

Figure 9.3: Trajectory estimation error over time. The solid line corresponds to dead-reckoning error, while the dashed line denotes the PHD-SLAM 2.0 trajectory error with radar measurements.

Figure 9.4 shows the mean OSPA distance between the estimated and actual landmark sets as a function of time. The OSPA distance is initially high when only a few measurements have been processed, then decreases and stabilizes at a relatively low value. Toward the later time steps, however, accumulated pose estimation error leads to a gradual increase in landmark localization error.

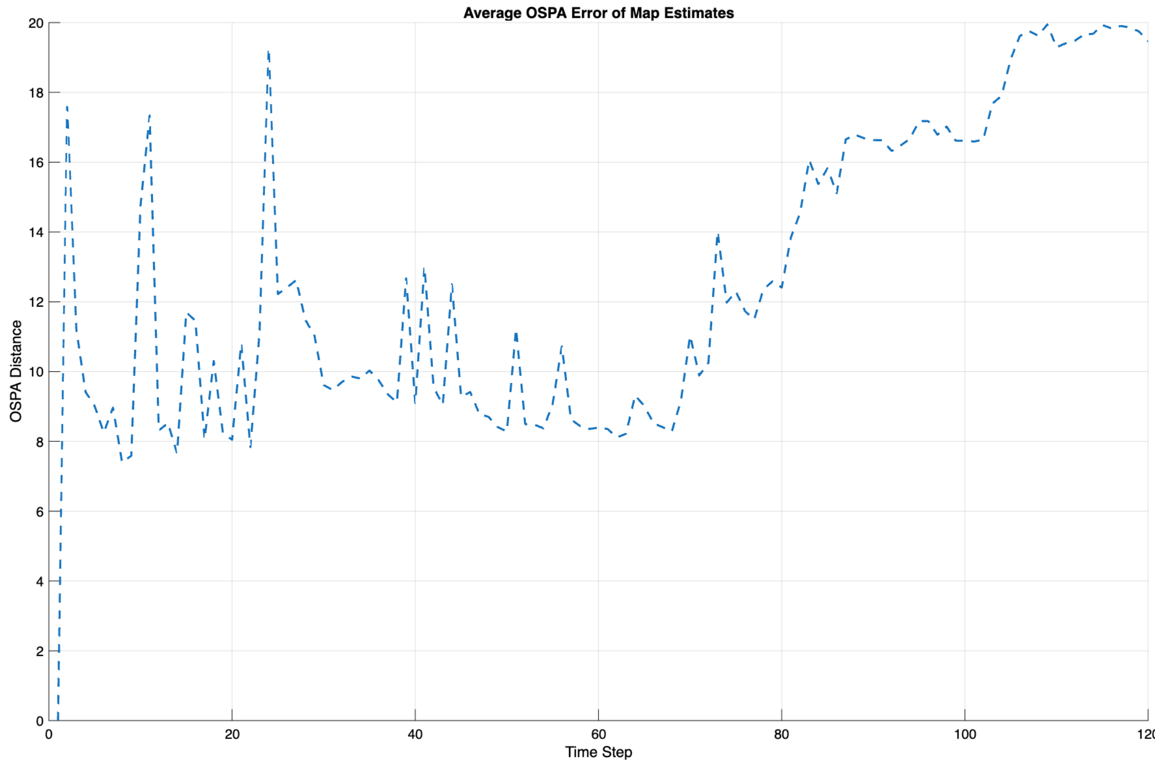

Figure 9.4: Mean OSPA distance for landmark estimates as a function of time.

This experiment demonstrates that PHD-SLAM 2.0 algorithm provides a viable and effective solution for radar-based SLAM, maintaining consistent performance under the synthetic scenario. The OSPA distances maintained a modest error rate and the estimated trajectory was substantially more accurate than the traditional dead reckoning approach. In addition, it confirms the feasibility of the data pipeline between the Stone Soup simulation environment and the MATLAB based PHD-SLAM 2.0, establishing a framework for future analysis and development.

Chapter 10

Conclusions and Future Work

10.1 Conclusion

This thesis presented a radar-centered approach to multi-object tracking from moving vehicle platforms, emphasizing the application of Random Finite Set (RFS) theory and the Gaussian Mixture Probability Hypothesis Density (GM-PHD) filter to realistic maritime scenarios. Through the use of datasets from the German Aerospace Center (DLR), the MANV, DARC, and DAAN collections, an analysis was conducted to evaluate the behavior of the GM-PHD filter under cluttered and dynamic conditions. The study demonstrated the importance of preprocessing radar detections using density-based clustering algorithms such as DBSCAN and HDBSCAN, which effectively group returns from the same physical target. These preprocessing steps were shown to be crucial for stable track consistency and performance evaluation in maritime radar scenes.

A complete GM-PHD implementation was used within the Stone Soup Python framework to enable repeatable experimentation and parameter analysis. The performance of the tracker was quantitatively assessed using the root mean square error (RMSE) and Optimal Subpattern Assignment (OSPA) metrics, providing an objective evaluation of localization accuracy and cardinality consistency. Parameter sensitivity studies revealed the critical role of pruning, merging, and birth thresholds in balancing track persistence with false alarm suppression. From these experiments, it was observed that while the GM-PHD filter can robustly maintain tracks in moderate clutter, its performance degrades in highly dynamic environments without adaptive parameter tuning. Finally, an initial exploration of the PHD-SLAM concept was conducted, outlining how RFS-based formulations can serve as a mathematical foundation for both target tracking and environment mapping. This connection illustrates the potential of leveraging radar as both a perception and mapping sensor for autonomous maritime platforms.

10.2 Future Works

While this research establishes a foundational framework for radar-based multi-object tracking using the GM-PHD filter, several promising areas remain open for further exploration. Future efforts should focus on adaptive clustering and measurement processing, in which the parameters of DBSCAN or HDBSCAN are dynamically tuned based on environmental conditions, sensor noise, or local target density to improve data-association consistency.

An especially compelling extension lies in the integration of PHD filtering with SLAM-based mapping. Using the probabilistic framework of the GM-PHD filter, it may be possible to identify and remove dynamically moving objects from the SLAM map in real time, ensuring that only stationary features contribute to the long-term environmental representation. This fusion of multi-object tracking and mapping would enable radar-based systems to simultaneously detect, track, and refine static landmark maps without interference from moving targets. Furthermore, incorporating other sensing modalities, such as optical cameras or LiDAR, could further improve target classification and reduce false-alarm rates. Finally, extending the current framework to operate on diverse datasets and sensing platforms would provide broader validation of its generality and adaptability. Collectively, these future developments will advance the realization of fully radar-centered perception systems capable of robust operation in complex and dynamic maritime environments.

Bibliography

[1] J. S. Fowdur, M. Baum, and F. Heymann, "Real-World Marine Radar Datasets for Evaluating Target Tracking Methods," *Sensors (Basel, Switzerland)*, vol. 21, no. 14, p. 4641, 2021. [Online]. Available: https://doi.org/10.3390/s21144641

[2] S. Hiscocks, J. Barr, N. Perree, J. Wright, H. Pritchett, O. Rosoman, M. Harris, R. Gorman, S. Pike, P. Carniglia, L. Vladimirov, and B. Oakes, "Stone Soup: No Longer Just an Appetiser," in *2023 26th International Conference on Information Fusion (FUSION)*, 2023.

[3] B.-T. Vo and W.-K. Ma, "The gaussian mixture probability hypothesis density filter," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4091–4104, 2006.

[4] J. Vargas, S. Alsweiss, O. Toker, R. Razdan, and J. Santos, "An overview of autonomous vehicles sensors and their vulnerability to weather conditions," *Sensors*, vol. 21, no. 16, p. 5397, 2021.

[5] A. Srivastav and S. Mandal, "Radars for autonomous driving: A review of deep learning methods and challenges," *IEEE Access*, vol. 11, pp. 97 147–97 168, 2023.

[6] J. Mullane, B.-N. Vo, M. D. Adams, and B.-T. Vo, "A random-finite-set approach to bayesian slam," *IEEE Transactions on Robotics*, vol. 27, no. 2, pp. 268–282, 2011.

[7] K. Granstrom, C. Lundquist, and O. Orguner, "Extended target tracking using a gaussian-mixture phd filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 4, pp. 3268–3286, 2012.

[8] M. Beard, S. Reuter, K. Granström, B.-T. Vo, B.-N. Vo, and A. Scheel, "Multiple extended target tracking with labeled random finite sets," *IEEE Transactions on Signal Processing*, vol. 64, no. 7, pp. 1638–1653, 2016.

[9] Y. Li, H. Xiao, Z. Song, R. Hu, and H. Fan, "A new multiple extended target tracking algorithm using phd filter," *Signal Processing*, vol. 93, no. 12, pp. 3578–3588, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168413001928

[10] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *IEEE Robotics Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[11] T. Bailey and H. Durrant-Whyte, “Simultaneous localization and mapping (slam): part II,” *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, September 2006.

[12] R. C. Smith and P. Cheeseman, “On the representation and estimation of spatial uncertainty,” *The international journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.

[13] J. Leonard and H. Durrant-Whyte, “Simultaneous map building and localisation for an autonomous mobile robot,” *Proc. IEEE Int. Workshop Intell. Robots Syst. (IROS)*, pp. 1442–1447, 1991.

[14] V. S. Chernyak and I. Y. Immoreev, “A brief history of radar,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 9, pp. 3–12, 2005.

[15] M. I. Skolnik, *Radar Handbook*, 3rd ed. New York: McGraw-Hill Education, 2008.

[16] M. A. Richards, *Fundamentals of Radar Signal Processing*, 2nd ed. New York: McGraw-Hill Education, 2014, copyright © 2014 by McGraw-Hill Education. All rights reserved.

[17] S. P. Lloyd, “Least squares quantization in PCM,” *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, March 1982.

[18] J. B. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1. Berkeley, CA, USA: University of California Press, 1967, pp. 281–297.

[19] 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 (KDD)*, ser. KDD’96. AAAI Press, 1996, pp. 226–231.

[20] R. J. G. B. Campello, D. Moulavi, and J. Sander, “Density-based clustering based on hierarchical density estimates,” *Advances in Knowledge Discovery and Data Mining*, vol. 7819, pp. 160–172, 2013.

[21] 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.

[22] G. Welch, G. Bishop *et al.*, “An introduction to the kalman filter,” 1995.

[23] J. R. Norris, *Markov chains.* Cambridge University press, 1998, no. 2.

[24] 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.

[25] B.-T. Vo, B.-N. Vo, and A. Cantoni, "Bayesian filtering with random finite set observations," *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1313–1326, 2008.

[26] J. Wishart, "The Generalised Product Moment Distribution in Samples from a Normal Multivariate Population," *Biometrika*, vol. 20A, no. 1-2, pp. 32–52, 1928. [Online]. Available: https://doi.org/10.1093/biomet/20A.1-2.32

[27] K. Granström, A. Natale, P. Braca, G. Ludeno, and F. Serafino, "Gamma gaussian inverse wishart probability hypothesis density for extended target tracking using x-band marine radar data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 12, pp. 6617–6631, 2015.

[28] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The mahalanobis distance," *Chemometrics and intelligent laboratory systems*, vol. 50, no. 1, pp. 1–18, 2000.

[29] L. Gao, G. Battistelli, and L. Chisci, "Phd-slam 2.0: Efficient slam in the presence of missdetections and clutter," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1834–1843, 2021.

[30] J. Mullane, B.-N. Vo, M. Adams, and B.-T. Vo, *Random Finite Sets for Robot Mapping SLAM.* Springer Berlin, Heidelberg, 2011. [Online]. Available: https://doi.org/10.1007/978-3-642-21390-8