# From 4D Radar Point-Cloud to State Estimation: A Comparison of Algorithms

Roan van der Voort

Delft University of Technology

# From 4D Radar Point-Cloud to State Estimation: A Comparison of Algorithms

by

Roan van der Voort

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Thursday September 18, 2025 at 15:00 PM.

| | | |
|---|---|---|
| Student number: | 4646452 | |
| Project duration: | May 1, 2025 – Septemeber 18, 2025 | |
| Thesis committee: | Prof. dr. ir. R. Babuska, | TU Delft, Cognitive Robotics |
| | Prof. dr. ir. H. Ceasar, | TU Delft, Cognitive Robotics |
| | Ir. A. Tasoglou, | Damen Research Development & Innovation |

Cover: IJ during Sail 2015. Modified from a photo by Christian Müller (own work). Licensed under Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)

TU Delft

# Preface

The completion of this Master’s thesis marks the culmination of a significant period of research and personal development. This journey would not have been possible without the guidance and support of several key individuals to whom I am deeply indebted.

I would first like to extend my sincerest gratitude to my university supervisor, Robert Babuska, and my company supervisors, Dimitrios Kotiadis and Athansios Tasoglou. Their guidance has been invaluable throughout this project. I am immensely grateful for the time they dedicated to reading my work, their insightful advice, and the constructive feedback that challenged me to think critically. Their willingness to engage in thoughtful discussions was instrumental in shaping the final results of this thesis. A special note of thanks is also due to Athansios Tasoglou for his crucial, hands-on assistance. His expertise in finding a viable configuration for the sensors was pivotal in allowing me to gather the real-world data that forms the foundation of this research.

The subject of this thesis lies at the intersection of two long-held interests. From a young age, I have been fascinated by the maritime world of ships, water, and vessels. As I grew older, this passion was complemented by a growing interest in the power of algorithms and their interaction with the physical world. This thesis provided a unique opportunity to explore that specific intersection, examining the intricate relationship between real-world maritime operations and algorithmic tracking. It has been a deeply rewarding experience to see these two distinct fields converge in this project.

On a personal note, I wish to thank my parents, family, and friends for their unwavering support and encouragement throughout this final capstone project. Their belief in me, especially during the more challenging moments, has been a constant source of motivation.

Finally, while the constraints of time and scope meant that not every possible avenue could be explored, nor the optimal algorithm definitively identified, I am proud of the work presented here. This thesis represents a thorough investigation into the topic, and it has been a challenging yet fulfilling conclusion to my Master’s degree.

*Roan van der Voort*
*Delft, September 2025*

# Summary

This work presents a comparative analysis of multi-target tracking algorithms for estimating the kinematic state and physical extent of maritime vessels from 4D millimetre-wave radar point clouds. The research addresses a gap in understanding the trade-offs between algorithmic complexity and motion model fidelity for inland maritime navigation.

The study evaluates three tracking paradigms, Tracking-by-Detection (TBD), the Probability Hypothesis Density (PHD) filter, and the Poisson Multi-Bernoulli Mixture (PMBM) filter, each tested with linear (Constant Velocity, Constant Acceleration) and non-linear (Constant Turn Rate) motion models. To represent vessel shape, three extended object models were implemented: the Random Matrix, Star-Convex Gaussian Process, and Principal Axis models. Performance was evaluated using a mix of simulated datasets with perfect ground truth and collected real-world datasets. A hyperparameter optimisation process was conducted for every algorithm combination using the Optuna framework, with the objective of minimising the Generalised Optimal Sub-Pattern Assignment metric using a specified cost function.

The results yielded several key, and at times counterintuitive, findings. Firstly, increased algorithmic complexity did not yield superior performance. The simpler TBD and PHD frameworks consistently outperformed the theoretically optimal but computationally intensive PMBM filter, which was prone to generating unstable tracks and excessive false positives. Secondly, non-linear motion models offered no significant advantage over the simpler linear models. This could be attributed to the slow dynamics of maritime vessels, the high sensor update rate, which makes linear extrapolation sufficient, and the fact that measurement noise often dominates any subtle gains from a more precise motion model. Thirdly, for extent estimation, the Random Matrix model demonstrated the best balance of accuracy, stability, and computational efficiency. The more complex Gaussian Process and Principal Axis models struggled with stability and practicality.

The study concludes that the optimal balance is achieved by pairing simple linear motion models with the Random Matrix extent model, yielding accurate cardinality estimation, reliable state tracking, and efficient computation. The findings underscore that robustness and interpretability outweigh algorithmic sophistication when designing maritime tracking systems based on 4D radar. Ultimately, the thesis establishes a principled performance baseline and highlights the diminishing returns of complexity in this domain.

# Contents

# Acronyms

**CA** Constant Acceleration. 3, 11, 14, 32, 34, 36, 44, 49
**CTRA** Constant Turn Rate and Acceleration. 3, 11, 12, 32, 44, 49
**CTRV** Constant Turn Rate and Velocity. 3, 11, 12, 32, 36, 44, 49
**CV** Constant Velocity. 2, 3, 11, 14, 32, 34, 36, 44, 49

**EKF** Extended Kalman Filter. 10, 11, 14, 23, 32, 36
**EOT** Extended Object Tracking. 3

**GIW** Gaussian Inverse Wishart. 12, 28, 29, 42, 44, 49
**GOSPA** Generalized Optimal Sub-Pattern Assignment. 25–29, 32, 46
**GP** Gaussian Process. 29

**KF** Kalman Filter. 2, 3

**MBM** Multi Bernoulli Mixture. 18–20
**MMW** millimetre wave. 1, 3, 5
**MOT** Multiple Object Tracking. 2

**PA** Principal Axis. 29, 32, 41, 44, 49
**PHD** Probability Hypothesis Density. 2, 3, 15–17, 19–22, 27, 28, 32, 36, 41, 42, 44, 46, 49
**PMBM** Poisson Multi-Bernoulli Mixture. 2, 3, 18–20, 22, 27–29, 42, 44, 46, 47, 49, 50
**PPP** Poisson Point Process. 18–20, 22

**RFS** Random Finite Sets. 2, 3, 15, 16, 18, 21–23, 46
**RM** Random Matrix. 28, 36, 41

**SCGP** Star Convex Gaussian Process. 12, 13, 28, 29, 41, 44, 45, 49

**TBD** Tracking-by-Detection. 2, 14, 21, 27–29, 36, 41, 42, 44, 49

**UKF** Unscented Kalman Filter. 10, 11, 14, 23, 32, 36

# 1 Introduction

The vision of fully autonomous ships navigating the world's waterways steadily progresses from concept to reality. The successful automation of commercial vessels, such as cargo ships and passenger ferries, promises to have a profound economic and safety impact on the maritime industry. However, significant technological hurdles must be overcome before the widespread adoption of such vessels becomes viable. Perception is a crucial step on this path; for any algorithm to make safe and reliable navigational decisions, it must have a complete and accurate model of the operational environment. Indeed, long before full automation is achieved, such an environmental model can be an invaluable decision-support tool to improve a captain's situational awareness.

A maritime perception system must operate reliably in a wide range of hostile conditions. It needs to be reliable whether it's dealing with intense sun glare on the water, thick fog, heavy rain, or the pitch-black of a rough sea at night. The system must perform with high fidelity across all these scenarios, providing both long-range foresight and high-resolution, close-quarters awareness.

Since no single sensor can reliably map the maritime environment on its own, a robust perception system has to blend the strengths of different sensors to make up for their individual weaknesses[1]. Conventional S-band and X-band marine radars, mandated for commercial vessels, offer excellent long-range detection capabilities largely unaffected by darkness or precipitation. However, their high mounting position creates a significant coverage gap in the near-field, and their relatively low resolution is insufficient for distinguishing between closely spaced objects, a critical requirement for safe manoeuvring in congested waters. Visual cameras provide rich, high-resolution data ideal for object classification, but are ineffective in darkness and severely hampered by fog, rain, and sun glare reflecting off the water's surface. Infrared cameras overcome the challenges of darkness by detecting thermal signatures, but struggle to differentiate between objects of similar temperature, and their performance is also degraded by humidity and precipitation. LiDAR, adapted from the automotive industry, generates high-resolution 3D point clouds ideal for precise object mapping. However, its reliance on laser light makes it highly vulnerable to signal attenuation and scattering from fog, rain, and sea spray, severely reducing its effective range and reliability in adverse weather, and it is prohibitively expensive when longer ranges are required.

In contrast, millimetre wave (MMW) radar offers a compelling alternative for near-field sensing. While its range is shorter than that of traditional marine radar, it provides a high-resolution point cloud. It remains robust across most weather and all lighting conditions, making it a prime candidate to fill the critical all-weather, close-quarters perception gap.

This thesis investigates the capabilities that one of these sensors, the 4D MMW radar, can contribute to a comprehensive maritime perception solution. The "4D" designation refers to the sensor's capacity to measure four specific parameters for every point in its generated cloud: range, azimuth, elevation, and Doppler velocity. It is therefore also referred to as a "3+1D" sensor, denoting three spatial dimensions and one velocity dimension. This sensor modality has undergone extensive development in recent years for the automotive industry, and its transfer to the maritime sector could provide a significant benefit, particularly in addressing the critical all-weather, near-field perception gap.

## 1.1. Tracking Challenges and Research Gap

Robust estimation techniques are essential for interpreting the rich, dense point clouds generated by 4D imaging radars. The core task is Multiple Object Tracking (MOT), which aims to estimate the number of objects in a scene and their individual states from a sequence of sensor measurements[2]. Essentially, MOT is an association problem: its goal is to match detections from different points in time to the same physical object, thereby maintaining a continuous and coherent trajectory. The benefit of this process is twofold. Firstly, it allows for the filtering of noisy sensor data to derive more accurate state estimates, such as an object's precise location. Secondly, it enables the inference of latent states, such as velocity and acceleration, which may not be directly observable from a single measurement but are critical for predicting future behaviour. Historically, when radar produced sparse and noisy detections, tracking algorithms were primarily used for this filtering and state estimation role, improving positional accuracy and predicting future trajectories from cluttered data. However, the rich, dense point clouds generated by modern 4D imaging radars, which can produce multiple detection points from an object's surface, have shifted the objective of tracking. Instead of tracking a single detection point per object, algorithms can now process multiple data points simultaneously. This evolution has given rise to the field of Extended Object Tracking (EOT), where the goal is not only to estimate an object's kinematic state but also its physical extent, such as its size and shape[3]. This is particularly crucial in maritime tracking, where resolving multiple reflections from large vessels into a single, coherent track with an accurate shape is essential for safe navigation and collision avoidance. To achieve this, an EOT algorithm must incorporate a shape model, with approaches ranging from simple parametric shapes like ellipses and rectangles to more complex, probabilistic methods such as Random Matrix models or non-parametric approaches like Gaussian Processes[4].

The problem of MOT is not unique to the maritime sector; it is a central challenge in aerospace, defence, and particularly in the automotive industry. Each domain presents distinct challenges: aerospace contends with high-speed targets, while automotive applications demand low-latency processing for dynamic traffic in cluttered urban environments. The extensive research in the automotive sector, driven by the development of self-driving cars, provides a valuable foundation. However, it has also led to a divergence in methodologies. The availability of vast, well-annotated public datasets has spurred the development of powerful, learning-based detectors. Consequently, the dominant paradigm in this field has become Tracking-by-Detection (TBD), which often pairs these advanced detectors with simpler, computationally efficient filters, such as the Kalman Filter (KF) with a linear motion model, to achieve real-time performance[5].

In contrast, other domains have continued to pursue more algorithmically advanced probabilistic filters to achieve the highest possible accuracy, especially when sensor data is less frequent or object dynamics are more complex. This includes sophisticated methods based on Random Finite Sets (RFS), such as the Probability Hypothesis Density (PHD) filter and, more recently, the Poisson Multi-Bernoulli Mixture (PMBM) filter. These methods provide a rigorous, unified Bayesian framework for handling detection uncertainty, clutter, and track management, but are often more computationally intensive[6].

This methodological difference presents a challenge when applying 4D radar technology to the maritime domain, where unique operational characteristics limit the direct transfer of existing solutions. The data-driven TBD paradigm, so successful in the automotive sector, is currently unfeasible due to the lack of large-scale, public maritime datasets containing the long, contiguous 4D radar sequences necessary for training robust detectors and evaluating trackers[7]. Simultaneously, the performance of advanced probabilistic filters is fundamentally constrained by the unique dynamics of marine vessels. Ships exhibit significant inertia and are subject to complex hydrodynamic forces, rendering their manoeuvres, such as turning, slow and inherently non-linear. The assumption that a simple linear motion model, for instance Constant Velocity

(CV), can accurately represent a vessel during a prolonged turn is therefore questionable. Such a model mismatch can lead to significant prediction errors, causing track divergence and incorrect identity assignments. Despite the theoretical robustness of advanced filters like the PMBM, their effectiveness is intrinsically linked to the fidelity of the underlying motion model. A distinct research gap therefore exists in the systematic investigation of the trade-offs between filter complexity and motion model fidelity within the specific context of maritime navigation using 4D radar data.

## 1.2. Thesis Aim and Research Questions

This thesis addresses the identified research gap by investigating the balance between algorithmic complexity and motion model fidelity for multi-object tracking in the inland maritime domain. The central aim is to determine the capabilities of state-of-the-art, purely algorithmic tracking methods when applied to the unique challenges presented by 4D MMW radar data in this environment. By shifting the focus away from data-driven paradigms prevalent in automotive research, this work concentrates exclusively on the capabilities and limitations of advanced probabilistic frameworks.

The core of this research is a comparative study evaluating the performance of advanced RFS-based tracking frameworks—specifically the Probability Hypothesis Density (PHD) filter and the Poisson Multi-Bernoulli Mixture (PMBM) filter—against a more traditional baseline tracker. These frameworks will be implemented with a range of motion models, including both linear (Constant Velocity, Constant Acceleration) and non-linear (Constant Turn Rate and Velocity/Acceleration) models, which are theoretically better suited to describing vessel manoeuvres. As a key component of EOT, the study will also explore the impact of different shape estimation models, such as the random matrix model or Gaussian Processes, on the ability to accurately represent the physical size and shape of diverse maritime objects. Ultimately, this work aims to provide clear insights into effective tracking architectures for marine automation when relying on principled, model-based estimation.

To provide a clear structure for this investigation and to ensure a focused analysis, the research is guided by the following primary questions and sub-questions:

1. **How do different motion models and state estimation filters influence the performance of multi-object tracking algorithms for inland maritime vessels using 4D radar point clouds?**
   - To what extent do non-linear motion models (e.g., CTRV, CTRA) improve tracking accuracy over simpler linear models (e.g., CV, CA) for representing vessel dynamics?
   - What is the performance trade-off between computationally efficient filters (like KF) and more complex or non-linear filters in the maritime context?
   - How does the inclusion of 4D data (elevation and Doppler velocity) and different measurement models (Cartesian vs. Polar) affect the stability and accuracy of these filter-model combinations?
2. **What is the optimal balance between algorithmic complexity and motion model fidelity for robustly tracking and estimating the physical extent of diverse objects in the inland maritime environment with 4D radar data?**
   - How do advanced probabilistic tracking frameworks (like PHD and PMBM) compare against each other and a more traditional tracking algorithm?
   - What is the impact of different extended object models (e.g., random matrix, multiple random matrices) on the ability to accurately estimate the size and shape of vessels with varying dimensions?

## 1.3. Scope of the Thesis

This thesis undertakes a systematic, comparative analysis of advanced probabilistic algorithms for multi-object tracking, specifically focused on the inland maritime domain. The research focuses on the performance trade-offs between algorithmic structures and their internal models when applied to data from 4D millimetre-wave radar. The goal is to provide a clear, evidence-based performance baseline for these tracking frameworks in a context that has not yet been extensively studied.

The core of the investigation is a comparison between three distinct filtering paradigms, chosen to represent a clear progression in tracking philosophy:

- **Tracking-by-Detection (TBD):** Serves as a practical baseline, reflecting the dominant paradigm in established fields like automotive research, where simpler filters are often paired with powerful detectors.
- **Probability Hypothesis Density (PHD) filter:** Investigated as a foundational Random Finite Set (RFS) approach. The PHD filter is a computationally efficient method that propagates the first-order moment of the multi-target distribution, making it a robust and well-established entry point into RFS tracking.
- **Poisson Multi-Bernoulli Mixture (PMBM) filter:** Represents the current state-of-the-art in RFS theory. While more computationally complex, the PMBM framework provides a more principled approach by maintaining distinct track hypotheses and probabilities of existence, allowing for higher-fidelity state estimation.

To thoroughly evaluate these frameworks, the study will methodically vary the underlying models for physical extent and kinematics. The evaluation will include:

- **Three extended object (shape) models:**
  - The Random Matrix model, which probabilistically describes an object's extent and is relatively simple to integrate into Bayesian frameworks.
  - The Gaussian Process model, chosen for its flexibility in representing more arbitrary, non-parametric shapes.
  - The Elliptical principal axis model, a parametric approach that directly estimates an object's orientation and axes lengths.
- **Four kinematic (motion) models:** Constant Velocity (CV), Constant Acceleration (CA), Constant Turn Rate and Velocity (CTRV), and Constant Turn Rate and Acceleration (CTRA) are chosen to assess the trade-off between simple linear models and more complex non-linear models that can better represent vessel manoeuvres.

This research is strictly confined to the maritime domain, with evaluation performed on a limited set of collected experimental scenes. These scenes are captured from a static sensor platform to isolate the performance of the tracking algorithms from the complexities of ego-motion compensation. The scenarios are designed to reflect key challenges in inland waterways and include a variety of situations, such as the tracking of single and multiple vessels of varying sizes.

The performance of each algorithm configuration will be evaluated based on three primary criteria:

- **Cardinality Accuracy:** The ability to correctly estimate the number of objects present in the scene.
- **State Estimation Accuracy:** The precision of the estimated kinematic states (position, velocity, orientation).
- **Shape Estimation Accuracy:** The fidelity of the estimated physical extent, measured against the ground truth bounding box.

To quantify these aspects, this research will primarily utilise the Optimal Sub-Pattern Assignment (OSPA) metric and its generalised variants (GOSPA). This family of metrics is specifically chosen for its ability to provide a single, rigorous cost that jointly penalises errors in target count, localisation, and state, making it exceptionally well-suited to the stated research focus.

Finally, it is important to reiterate that this work is exclusively focused on algorithmic, model-based approaches and does not involve machine learning. The primary goal is a high-level comparative analysis of the chosen filters and models, rather than exhaustive fine-tuning of any single algorithm.

## 1.4. Thesis Structure

The remainder of this thesis is structured to guide the reader logically through the research, from foundational concepts to conclusions.

Chapter 2 establishes the theoretical foundation for this work. It details the operating principles of 4D MMW radar, the mathematics of Bayesian state estimation, and the specifics of the chosen motion and extent models. This chapter also provides a thorough explanation of the theoretical underpinnings of the various tracking frameworks that form the core of the investigation.

Following this, Chapter 3 outlines the methodology, detailing the practical implementation of the tracking algorithms. It describes the design of the simulation environment, the process for generating data for the defined maritime scenarios, and the specific evaluation metrics chosen for the comparative study.

Chapter 4 presents the quantitative results from the simulation experiments, directly comparing the performance of each tracker configuration across the defined scenarios and metrics. These results are then analysed and interpreted in Chapter 5, which provides an in-depth discussion of their implications and explicitly answers the research questions posed in this introduction.

Finally, Chapter 6 concludes the thesis by summarising the key findings and contributions, acknowledging the limitations of the research, and providing concrete recommendations for future work in this area.

# 2
# Background and Theory

This chapter lays the theoretical groundwork for the multi-object tracking methods studied in this thesis. We begin with the operating principles of the 4D millimeter-wave radar that provides the measurements. We then review Bayesian state estimation, the mathematical basis of the tracking algorithms. Next, we present the models used to describe target kinematics and physical extent. Finally, we introduce the three tracking paradigms compared in this work: classical tracking-by-detection, the Random Finite Set-based Probability Hypothesis Density filter, and the Poisson multi-Bernoulli mixture filter.

## 2.1. Principles of 4D Millimetre-Wave (MMW) Radar

This section breaks down the fundamentals of 4D millimetre-wave (mmWave) imaging radar, a key technology for understanding the environment around a system. The radar's hardware and how it processes signals directly shape the data it produces. This, in turn, influences the design of the tracking algorithms we'll discuss later.

### 2.1.1. Operating Principles

Millimetre-wave radar is a reliable sensor for demanding environments. These systems typically operate in the W-band (76–81 GHz)[8]. Compared with optical sensors (cameras, LiDAR), mmWave frequencies are much less affected by fog, rain, snow or poor lighting, so the radar continues to deliver range, angle and velocity measurements when visibility is degraded. This resilience makes mmWave radar well suited to safety-critical tasks such as maritime automation.

### 2.1.2. The Four Dimensions

The name "4D" (or occasionally "3+1D") radar is derived from its capability to measure four distinct properties for each detected point.

The primary and most fundamental dimension measured is range, which quantifies the distance from the sensor to a target. This measurement is precisely calculated based on the time-of-flight of the transmitted electromagnetic wave. The accuracy and resolution of this measurement are directly proportional to the system's operational bandwidth and a trade-off between range and Doppler resolution. For the sensor under study the range resolution is around $0.65$ m and a maximum operational range of approximately $250$ m.[8]

The angular position of a target within the sensor's field of view is described by its azimuth (horizontal angle) and elevation (vertical angle). Both angular dimensions are determined through the processing of signals received by the radar's antenna array. Modern systems often employ architectures like multiple transmit and receive antennas to synthesise a larger virtual aperture. This configuration yields an approximate azimuth resolution of $15°$. The elevation resolution however is much worse being $60°$.

The final dimension is the Doppler velocity, which quantifies the radial speed of a target, indicating its movement directly towards or away from the sensor. This is calculated by detecting the frequency shift in the returning electromagnetic wave, known as the Doppler effect. The system is engineered to resolve velocity differences as minute as $0.5$ m/s, providing indispensable information for the robust tracking and prediction of dynamic objects' trajectories.

To illustrate how these limits impact the structure of the final point cloud, consider a vessel 100 m away which can produce detection uniformly across its surface. This is illustrated in Figure 2.1. The azimuth resolution of $15°$ at 100 m translates to a lateral resolution of approximately 26 m, meaning that features on the vessel closer than this distance side-to-side may not be separately resolved or jump around shown in the second graph of Figure 2.1. The range resolution of 0.65 m means that any two points on the vessel closer than this distance along the line of sight will be indistinguishable in range. With a similar binning effect also happening to the angles in this specific sensor. Leading to the final graph in Figure 2.1. Finally, the velocity resolution of $0.5$ m/s allows for distinguishing between objects moving at slightly different speeds, which is crucial for tracking multiple targets in motion.

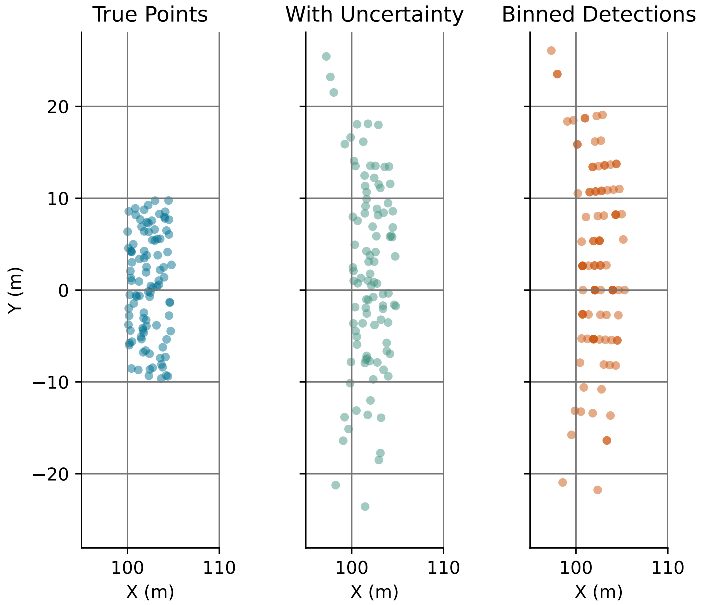


**Figure 2.1:** Example illustration of range/angle bins and the associated measurement uncertainty for the 4D mmWave radar.

### 2.1.3. Point Cloud Characteristics and Limitations

The final output from the radar is a point cloud, which is a collection of individual detection points in 3D space, each with its own velocity reading. A key thing to understand about radar point clouds is that they are naturally sparse and unstructured, as can be seen from the Single frame graph in Figure 2.2. This is quite different from the dense and orderly point clouds you'd get from a LiDAR sensor. Because of this sparsity, objects are often represented by just a few points, making it difficult to work out their exact shape or size from a single snapshot.

On top of that, the data is often noisy. It can be affected by clutter, which are unwanted echoes from things in the environment like sea spray or rain, or simply random noise present in the signal. This effect is visible when accumulating across multiple frames, for example in the 10 Frames graph of Figure 2.2. The sparse points farther away are clutter and don't correspond to any real object. Another major issue is multipath reflections, where the radar signal bounces off multiple surfaces like the water and then a ship's hull before returning. These reflections can create 'ghost' targets that aren't really there or make real objects appear in the wrong place. Such a ghost target is for example visible in Figure 3.3d, the ofshoot from the real target not being there in reality.

Together, these issues of sparsity, noise, clutter, and multipath reflections make it a real challenge to design algorithms that can reliably detect and track objects. The coarse angular resolution also makes it hard to distinguish between objects that are close together. As a result, we need clever data processing and filtering techniques to pull reliable information from the radar's point cloud and build an accurate picture of the world.

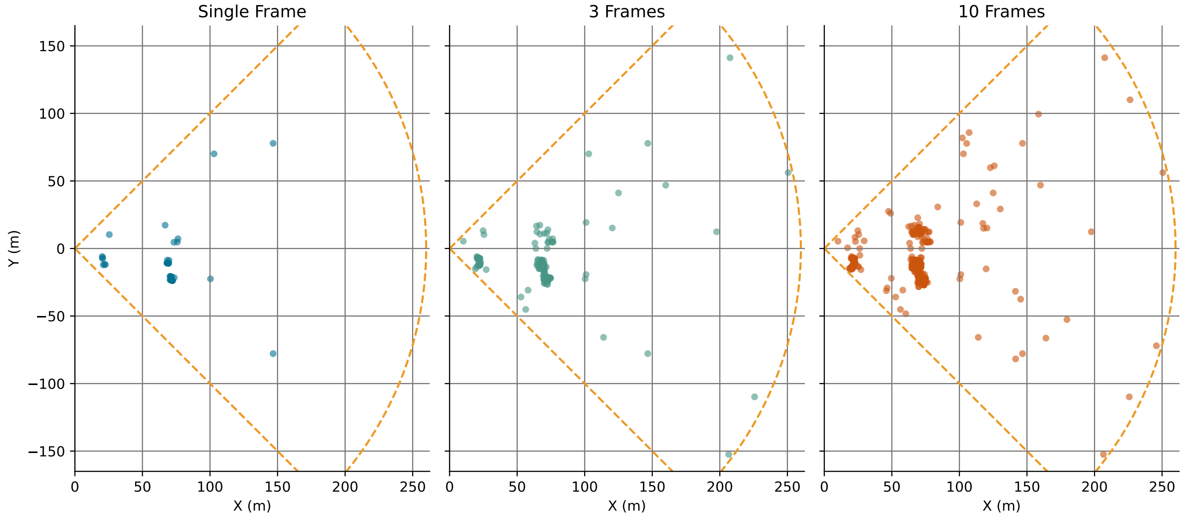


**Figure 2.2:** Schematic of the 4D mmWave radar point cloud structure. Each detection provides a 3D position (x,y,z) and an associated radial velocity; measurement uncertainties from range, azimuth and elevation produce binned, sparse detections and angular/range uncertainty cones.

## 2.2. Fundamentals of Bayesian State Estimation

Fundamentally, target tracking is an estimation problem: we must infer the true, unobserved state of a dynamic object—such as its position, velocity, and orientation—from a sequence of indirect measurements. With multi target tracking only adding more complications to the estimate. Such as estimating a varying number of objects, that maybe obstructing each other and may not be governed by the same models (differing size, speed or just static objects). This section outlines the core probabilistic framework for solving this problem, known as recursive Bayesian estimation, and introduces its most common implementations: the Kalman Filter and its non-linear extensions.

### 2.2.1. The State Estimation Problem

A dynamic system, such as a single moving target, can be characterised at any discrete time step, $k$, by a state vector, denoted $\boldsymbol{x}_k$. This vector contains all the necessary information to describe the object's physical properties of interest. For example, in a two-dimensional plane, the state might be represented as $\boldsymbol{x}_k = [p_x, p_y, v_x, v_y]^\top$, capturing the position and velocity components along the x and y axes.

In almost any practical scenario, this state vector cannot be observed directly. Instead, we obtain a sequence of measurements, $\boldsymbol{z}_k$, from sensors. These measurements are typically a function of the true state but are always distorted by sensor noise. The central challenge, therefore, is to filter out this noise and derive the most accurate possible estimate of the state $\boldsymbol{x}_k$ using the sequence of all available measurements up to that time, $\boldsymbol{Z}_k = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k\}$.

This process is inherently probabilistic. We do not seek a single point estimate but rather aim to determine the complete posterior probability density function (PDF), $p(\boldsymbol{x}_k|\boldsymbol{Z}_k)$, which encapsulates all statistical information about the current state, given all the evidence received so far. In the context of this work this PDF is chosen as a Gaussian for the kinematic state, and varying for the extent state.

### 2.2.2. The Recursive Bayesian Filter (Bayes Filter)

The Bayes filter offers a recursive, probabilistic approach to this estimation problem, cycling between two key steps: prediction and update. Assuming the posterior PDF at time $k-1$, $p(\boldsymbol{x}_{k-1}|\boldsymbol{Z}_{k-1})$, is known, the filter computes the posterior PDF at time $k$, $p(\boldsymbol{x}_k|\boldsymbol{Z}_k)$, when a new measurement, $\boldsymbol{z}_k$, arrives. A simple diagram of this cycle is shown in Figure 2.3.

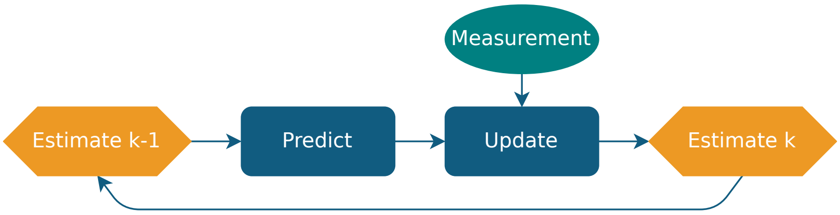


**Figure 2.3:** The Bayesian filter predict–update cycle. The posterior at time $k-1$ is predicted forward using the process model to form the prior at time $k$, which is then updated with the measurement at time $k$ to produce the posterior at time $k$.

This recursive process is built upon two key models:

1. A **process model** (or motion model), described by the transitional density $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$, which defines how the state evolves from one time step to the next.
2. A **measurement model**, described by the likelihood function $p(\boldsymbol{z}_k|\boldsymbol{x}_k)$, which relates the noisy measurement to the true state at the current time step.

#### Prediction Step

The prediction step, also known as the time update, uses the process model to predict the state PDF at time $k$, given all measurements up to time $k-1$. This predicted PDF, $p(\boldsymbol{x}_k|\boldsymbol{Z}_{k-1})$, is known as the *prior* for time step $k$. It is computed by marginalising over all possible states at the previous time step, $\boldsymbol{x}_{k-1}$:

$$p(\boldsymbol{x}_k|\boldsymbol{Z}_{k-1}) = \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})p(\boldsymbol{x}_{k-1}|\boldsymbol{Z}_{k-1})d\boldsymbol{x}_{k-1} \tag{2.1}$$

This integral propagates the state distribution forward in time, incorporating the uncertainty associated with the system's dynamics.

#### Update Step

The update step, or measurement update, refines the predicted (prior) state distribution using the latest measurement, $\boldsymbol{z}_k$. This is achieved using Bayes' theorem to compute the *posterior* state distribution, $p(\boldsymbol{x}_k|\boldsymbol{Z}_k)$:

$$p(\boldsymbol{x}_k|\boldsymbol{Z}_k) = \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{Z}_{k-1})}{p(\boldsymbol{z}_k|\boldsymbol{Z}_{k-1})} \tag{2.2}$$

Here:

- $p(\boldsymbol{z}_k|\boldsymbol{x}_k)$ is the *likelihood* of the measurement given the state, provided by the measurement model.
- $p(\boldsymbol{x}_k|\boldsymbol{Z}_{k-1})$ is the *prior* distribution, calculated in the prediction step.
- $p(\boldsymbol{z}_k|\boldsymbol{Z}_{k-1})$ is the marginal likelihood or *evidence* of the new measurement. It acts as a normalisation constant and is calculated as: $p(\boldsymbol{z}_k|\boldsymbol{Z}_{k-1}) = \int p(\boldsymbol{z}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{Z}_{k-1})d\boldsymbol{x}_k$.

The resulting posterior, $p(\boldsymbol{x}_k|\boldsymbol{Z}_k)$, represents our updated belief about the state and becomes the basis for the next prediction step.

### 2.2.3. Implementations of the Bayes Filter

While the Bayes filter offers a complete conceptual framework, the integrals involved in its prediction and update steps are often intractable for general non-linear or non-Gaussian models. Practical implementations, therefore, rely on specific assumptions that permit an analytical or approximate solution.

#### The Kalman Filter

The Kalman filter, first developed for discrete-time processes by Rudolf E. Kalman in 1960, is the optimal recursive Bayesian filter for a specific, and rather restrictive, set of circumstances. It provides a closed-form solution when:

1. **Linearity**: Both the process model and the measurement model are linear functions of the state.
2. **Gaussianity**: The initial state and the noise associated with both the process and the measurements are assumed to be Gaussian.

The general linear Gaussian state-space model can be expressed as:

- Process Model: $\boldsymbol{x}_k = \boldsymbol{F}_k \boldsymbol{x}_{k-1} + \boldsymbol{\zeta}_{k-1}$, with $\boldsymbol{\zeta}_{k-1} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Z}_{k-1})$
- Measurement Model: $\boldsymbol{z}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{\epsilon}_k$, with $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_k)$

Under these conditions, the posterior distribution $p(\boldsymbol{x}_k | \boldsymbol{Z}_k)$ remains Gaussian at every time step. The Kalman filter, therefore, does not need to compute the entire PDF but simply propagates its first two moments—the mean vector and the covariance matrix—through the prediction and update equations. For linear systems with Gaussian noise, the Kalman filter is the optimal estimator in the minimum mean-squared error sense. Its primary limitation, however, is its strict requirement for linearity, which is seldom met in real-world applications.

#### The Extended Kalman Filter

To address the Kalman Filter's strict linearity requirement, the Extended Kalman Filter (EKF) approximates non-linear models. It handles non-linearity by linearising the process and measurement models at each time step, using a first-order Taylor series expansion around the current state estimate. A non-linear system can be written generally as:

- Process Model: $\boldsymbol{x}_k = g(\boldsymbol{x}_{k-1}) + \boldsymbol{\zeta}_{k-1}$
- Measurement Model: $\boldsymbol{z}_k = f(\boldsymbol{x}_k) + \boldsymbol{\epsilon}_k$

The EKF approximates these functions with their Jacobians (matrices of first-order partial derivatives), effectively turning the non-linear problem into a time-varying linear one. The standard Kalman filter equations are then applied to this linearised approximation.

While the EKF has seen widespread use, its performance depends heavily on how well the linearisation reflects the true model. For highly non-linear systems, this first-order approximation can be poor, leading to significant errors and, in some cases, causing the filter to diverge entirely.

#### The Unscented Kalman Filter

The Unscented Kalman Filter (UKF) offers a more straightforward alternative for handling non-linear systems[9]. Rather than approximating the non-linear functions, as the EKF does, the UKF uses a deterministic sampling approach to approximate the probability distribution itself.

This technique, known as the unscented transform, represents the state distribution using a minimal set of carefully chosen sample points, called sigma points. These points are selected to precisely capture the mean and covariance of the distribution. Each sigma point is then propagated directly through the true non-linear functions, and a new mean and covariance are calculated from the resulting transformed points.

The UKF's real advantage over the EKF is its ability to bypass the often tricky and error-prone calculation of Jacobian matrices. By working with the true non-linear models, it generally provides a more accurate approximation of the posterior distribution (accurate to at least the second order for any non-linearity) and is less susceptible to the divergence issues that can plague the EKF in highly non-linear scenarios.

## 2.3. System Models for Maritime Tracking

As mentioned previously the multi-target tracking problem can be viewed as a state estimation problem. This state contains both the kinematic state (i.e. position, velocity, etc.) and the extent state (i.e. size, shape, orientation, etc.) of each object. To estimate these states a model is required to describe how these states evolve over time (motion model) and how the sensor measurements relate to these states (measurement model or extent model). This section describes the specific models used in this work for maritime vessel tracking using 4D radar point clouds.

### 2.3.1. Kinematic Motion Models

Kinematic motion models are a fundamental component of state estimation for tracking, providing a mathematical description of how an object's state is predicted to evolve over time. These models describe the motion of an object based on its position, velocity, and acceleration, without considering the underlying forces that cause the motion. They are integral to the prediction step within recursive filtering frameworks, such as the Kalman filter and its non-linear variants, where they are used to project the state estimate forward from one time step to the next. The choice of model represents a trade-off between fidelity to the true object dynamics and computational tractability. Deviations from the model's assumptions are typically accounted for by introducing process noise, which provides the filter with the flexibility to track behaviours that are not perfectly captured by the model dynamics.

#### Linear Models

Linear models are characterised by linear state transition equations, making them directly compatible with the standard Kalman filter. Their computational efficiency makes them a common choice, particularly for non-manoeuvring targets.

*Constant Velocity (*

The CV model is one of the simplest and most widely used kinematic models. It assumes that the object moves with a constant velocity vector between discrete time steps. While this assumption is rarely perfectly true, the model is effective for tracking objects moving in a relatively straight line or when the observation interval is short. It often serves as a baseline model in tracking systems due to its simplicity and low computational cost.

*Constant Acceleration*

The Constant Acceleration (CA) model extends the CV model by assuming the object moves with a constant acceleration vector. This allows the model to better represent objects that are systematically changing their velocity, such as vehicles accelerating or decelerating in a straight line. Although more descriptive than the CV model for such scenarios, it remains unable to effectively model turning manoeuvres and requires a higher-dimensional state vector [10].

#### Non-Linear Models

To address the limitations of linear models in tracking manoeuvring targets, non-linear models are employed. These models explicitly account for the rotational dynamics of an object and, due to their non-linear state transition functions, require the use of filters such as the Extended Kalman Filter or Unscented Kalman Filter.

*Constant Turn Rate and Velocity*

The Constant Turn Rate and Velocity (CTRV) model assumes the object moves with a constant speed and a constant rate of turn. This allows it to describe an object moving along a circular path, providing a significant improvement over linear models for tracking turning targets. However, its primary limitation is the assumption of constant speed, which is often unrealistic as vehicles typically decelerate before a turn and accelerate afterwards [10].

*Constant Turn Rate and Acceleration*

The Constant Turn Rate and Acceleration (CTRA) model enhances the CTRV model by incorporating a constant longitudinal acceleration term. This allows the object's speed to change linearly over the time interval whilst it is turning. By modelling simultaneous changes in speed and heading, the CTRA model offers a more comprehensive and realistic representation of vehicle manoeuvres, making it highly suitable for applications such as automotive and vessel tracking [11].

#### Relevance to Maritime Dynamics

The dynamics of marine vessels present unique challenges for tracking. Due to their enormous mass and the hydrodynamic forces they are subject to, vessels possess significant inertia. This results in slow, prolonged manoeuvres rather than the abrupt changes in direction often seen in automotive applications. Consequently, non-linear models that can describe these sustained changes are theoretically better suited for vessel tracking.

The assumptions of the CTRA model, which account for both a constant turn rate and a constant acceleration, align well with the typical behaviour of a vessel altering its course. During such a manoeuvre, a ship will have a non-zero rate of turn whilst potentially adjusting its engine power, resulting in a change in speed. This makes the CTRA model a more physically representative choice compared to simpler linear models or the constant-speed CTRV model for capturing the dynamics of a turning ship [11].

### 2.3.2. Measurement Models

Any specific sensor introduces noise into the data. Measurement models relate the true state of the system to the observed measurements, often incorporating noise characteristics. These can be linear or non-linear depending on the sensor. Linear models are often used for simplicity, but non-linear models can provide better accuracy in complex scenarios. The noise of this type of radar sensor is non-linear.

### 2.3.3. Extended Object Models

In the tracking of extended objects, a central element is the mathematical representation of the object's physical size and shape, referred to as its extent. The choice of model dictates how an object's extent is characterised and how new measurements are incorporated into the state estimation process. The extent model can also be viewed as a measurement model, as it defines how sensor measurements relate to the object's state. These models can be broadly categorised into parametric and probabilistic approaches. While there are more approaches only the background of the models used in this work are discussed here.

#### Probabilistic Models

Probabilistic models characterise the object's extent using a statistical distribution, which can capture not only the shape but also the inherent uncertainty in that shape.

*Random Matrix Model*

A prominent probabilistic approach is the Random Matrix Model, also commonly referred to as Gaussian Inverse Wishart (GIW), which characterises an object's extent as a symmetric positive definite random matrix[12], making the assumption that the measurements of an object are Gaussian distributed around its centroid. The state of the extent is therefore modelled using an inverse Wishart distribution as PDF instead of a Gaussian as for the kinematic state, which is the conjugate prior for the Gaussian-distributed sensor detections[13]. This formulation provides a mathematically elegant way to handle uncertainty in an elliptical shape within a Bayesian filtering framework. For non-ellipsoidal shapes, this model can be adapted by representing the object as a composite of multiple ellipsoidal sub-objects, each represented by its own random matrix[14].

For what this would look like in practice see Figure 2.4, which shows the estimated elliptical extent of a simple rectangle using the GIW random-matrix model, along with the detections used for the update.

*Star Convex Gaussian Process (GP) Model*

For objects with more complex or arbitrary shapes that cannot be well-approximated by an ellipse, a Star Convex Gaussian Process model offers a flexible, non-parametric alternative[15].

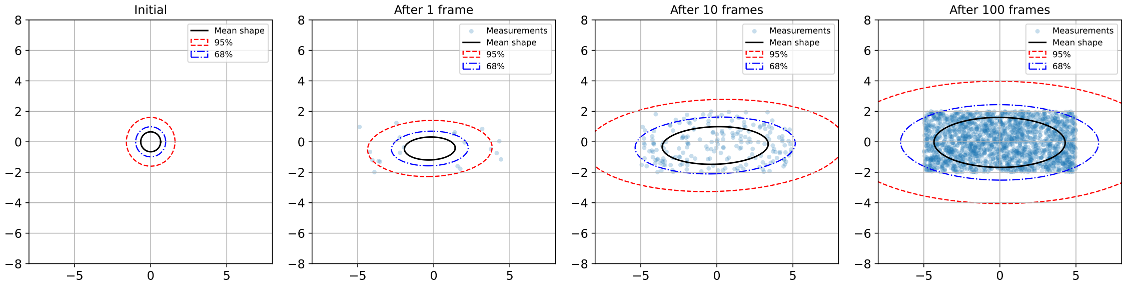


**Figure 2.4:** Random Matrix (IW) extent estimation: estimated elliptical extent and accumulated detections used for the update.

A Gaussian Process can be considered a distribution over functions. In this context, it is used to model the boundary of an unknown object, typically by learning a radial function that maps an angle to a radius from the object's centre[4]. This approach allows for the online learning of arbitrary star-convex shapes and naturally provides an analytical representation of the shape with well-defined confidence intervals, which is advantageous for gating and data association[15].

Figure 2.5 illustrates the SCGP extent estimator.

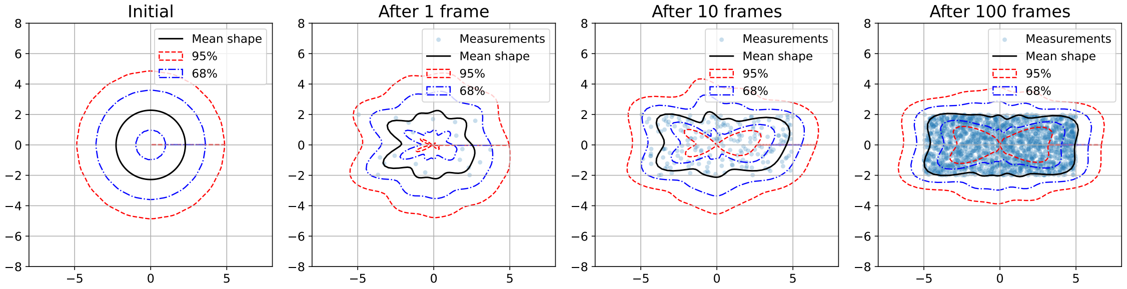


**Figure 2.5:** Star-Convex Gaussian Process shape estimation over time

### Parametric Models

Parametric models describe an object's shape using a finite and fixed number of parameters. These models are often favoured for their computational efficiency and simplicity.

*Elliptical Principal Axis Model:*

A more specific parametric approach is the principal-axes-based model, which directly estimates the ellipse's orientation and the lengths of its axes[16], [17]. This explicit parameterisation allows for individual uncertainty values to be assigned to each extent parameter, offering a flexible representation. This stands in contrast to some other methods where the uncertainty is limited to a single scalar value.

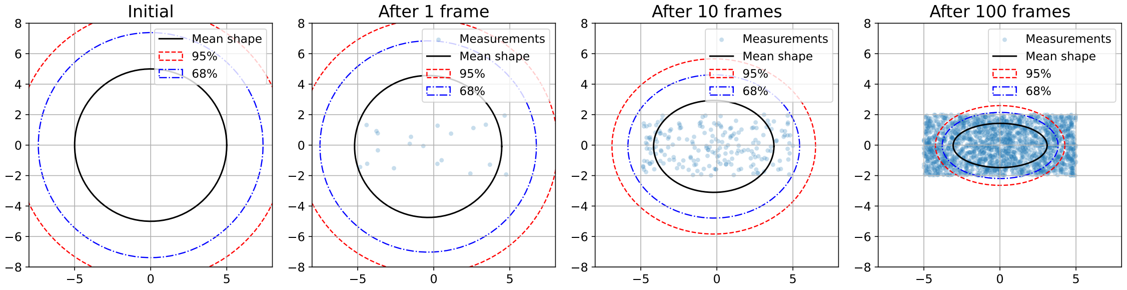


**Figure 2.6:** Principal Axis shape estimation

## 2.4. Multi-Target Tracking Frameworks

### 2.4.1. The Tracking-by-Detection Paradigm

The Tracking-by-Detection (TBD) paradigm is a dominant and conceptually straightforward approach to solving the Multiple Target Tracking problem. It is currently the most prevalent method employed by leading tracking algorithms, particularly within the automotive domain [2]. Effectively by separating the algorithm into separate steps it can otherwise be viewed as a sequential single target tracking problem [2].

#### Core Concept

The fundamental principle of TBD is the decoupling of the multi-object tracking challenge into two distinct, sequential steps: detection and data association [2]. In this framework, object detection is performed on a frame-by-frame basis, independent of the state estimation process. The subsequent association step then links these individual detections across time to form trajectories. This separation allows for the independent development and optimisation of both the detector and the tracker. Consequently, the complex multi-target problem is effectively reduced to a series of more manageable single-target tracking and assignment problems, where the primary goal is to match a new set of detections to existing tracks at each time step [4].

*Architecture*

A typical TBD architecture is modular, comprising several key components that work in sequence to process sensor data and output stable tracks. This structure, while allowing for significant variation in the complexity of each module, generally adheres to a consistent workflow [5].

*Detection*

The initial stage of the TBD pipeline is object detection. This component is responsible for processing the raw sensor input (e.g., a 4D radar point cloud) for each individual frame or scan and generating a set of object detections. These detections typically take the form of 3D bounding boxes with attributes such as position, dimensions, orientation, and classification [5]. But can also be create by a clustering algorithm clustering related points. The performance of the entire tracking system is highly dependent on the quality of these initial detections.

*Association*

Once detections are generated for a given frame, the association module attempts to match them with the existing tracks from previous frames. This is fundamentally an assignment problem. To achieve this, a similarity or cost metric is first computed between each new detection and each existing track. A widely used metric for this purpose is the Intersection over Union between the detection’s bounding box and the predicted bounding box of the track [18]. Other metrics, such as Mahalanobis distance or appearance-based feature similarity, can also be used [19]. Based on the resulting cost matrix, an assignment algorithm is employed to find the optimal set of matches. The Hungarian Algorithm is a common choice for this task as it finds the optimal solution that minimises the total cost [20].

*State Estimation*

For each successfully matched track-detection pair, the track’s state is updated. For unmatched tracks, a prediction is made. This is typically handled by a recursive Bayesian filter. The Kalman Filter, along with its variants like the Extended Kalman Filter or Unscented Kalman Filter, is the most common choice for this component [18]. The filter uses a motion model to predict the track’s state (e.g., position, velocity) at the current time step. When a new detection is associated with the track, the filter uses this measurement to correct the prediction and update the state estimate, thereby filtering out noise and providing a smoothed trajectory. Simple kinematic models, such as the Constant Velocity or Constant Acceleration models, are frequently employed due to their computational efficiency and sufficiency for short-term prediction in many scenarios [5].

*Track Management*

The final component is responsible for the lifecycle of the tracks. This involves a set of logical rules for initiating, maintaining, and terminating tracks. A new track is typically initiated for any detection that cannot be associated with an existing track. To prevent false positives from immediately creating persistent tracks, a probation period may be used, where a new track is only confirmed after it has been successfully associated with detections for a certain number of consecutive frames [5]. Conversely, a track that is not associated with any detection for a specified number of frames is deemed to have left the scene (or to be occluded) and is terminated [18]. This management logic is crucial for maintaining a clean and accurate set of current object tracks.

### 2.4.2. Random Finite Set Theory

#### Core Concept

The core concept of Random Finite Sets (RFS) theory, particularly in the context of multi-target tracking, is to model the entire collection of targets as a single random entity rather than treating each target individually [21]. This entity is a finite set whose elements—the individual target states—and whose number of elements (known as cardinality) are both random [21].

Essentially, RFS theory generalises the statistics of random vectors to the statistics of random sets [21]. In traditional multi-target tracking, the state of each target is typically modelled as a random vector, which complicates the handling of a fluctuating number of targets and the association of measurements to tracks [21]. In contrast, the RFS approach models the complete multi-target state as a single RFS and, likewise, models the set of measurements received from sensors as another RFS [21].

This provides a unified, top-down Bayesian framework that inherently accounts for the key uncertainties in multi-target tracking [21]:

- A time-varying number of targets (object birth and death) [21].
- Missed detections and false alarms (clutter) [21].
- The ambiguity of measurement origin [21].

By treating the collection of targets as a single entity, the RFS framework allows for the simultaneous estimation of both the number of targets and their respective states, naturally incorporating mechanisms for track initiation and termination without requiring separate data association steps [21].

### 2.4.3. The Probability Hypothesis Density Filter

The Probability Hypothesis Density (PHD) filter, first formalized by Mahler[22], [23], represents a significant advancement in multi-target tracking by providing a computationally tractable approximation to the optimal multi-target Bayesian filter [23]. The core innovation of the PHD filter is its departure from propagating the full multi-target posterior density, a process whose complexity grows exponentially with the number of targets and measurements. Instead, it propagates a lower-order statistical moment of this density, offering a practical solution to an otherwise intractable problem.

#### Core Concept

The PHD filter for extended targets operates within the mathematical framework of Random Finite Sets (RFS). This approach is essential for multi-target tracking as it circumvents the combinatorial complexity of traditional data association methods [21]. The core challenge in tracking is that the set of measurements received by a sensor has an uncertain origin, arising from an unknown number of targets and spurious clutter. By modelling the entire collection of target states as a single random finite set, the filter can handle a time-varying number of targets, missed detections, and false alarms in a unified and systematic way. This random finite set is a set of elements, for example Gaussian components representing a motion state, where the number of elements is itself a random variable.

For extended targets, objects that can generate multiple measurements per time step due to their significant physical size, this framework requires a more sophisticated state representation. The single-target state vector, $x$, must be augmented to include not only its kinematic properties (e.g., position, velocity) but also a description of its physical extent. This is commonly achieved by modelling the extent using simple geometric shapes (like an ellipse, defined by its semi-axes and orientation) or more advanced, flexible representations like Random Hypersurface Models [23], [24].

The central innovation of the PHD filter is that instead of propagating the full, computationally intractable multi-target posterior density, it propagates only its first-order statistical moment. This is the probability hypothesis density, or intensity function, $D(x)$, which exists over the augmented single-target state space (including extent). The integral of the PHD over any region of this state space yields the expected number of targets in that region, while the peaks of the intensity function correspond to the likely kinematic states and extents of the targets [24].

The filter operates through a recursive three-step process: prediction, update and prune/merge. This is summarised in Figure 2.7.

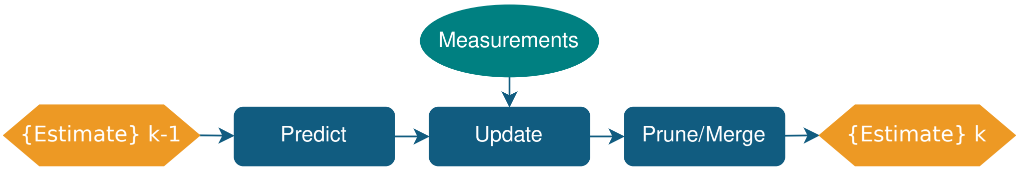


**Figure 2.7:** Schematic of the PHD filter recursion for extended targets: prediction (survival + birth), measurement partitioning, update of component weights via partition likelihoods and detection probability, and prune/merge.

*Prediction Step*
The prediction step evolves the intensity from the previous time step, $k-1$, to the current time step, $k$. This involves accounting for targets that survive from the previous state, as well as new targets that are born into the surveillance area. The predicted PHD, $D_{k|k-1}(x)$, is the sum of the intensity of surviving targets and the intensity of new born targets, $\gamma_k(x)$:

$$D_{k|k-1}(x) = \int p_{S,k}(\xi) f_{k|k-1}(x|\xi) D_{k-1|k-1}(\xi) d\xi + \gamma_k(x) \tag{2.3}$$

Here, the integral term represents the contribution from surviving targets:

- $D_{k-1|k-1}(\xi)$ is the posterior intensity from the previous time step.
- $p_{S,k}(\xi)$ is the probability that a target with state $\xi$ at time $k-1$ will survive to time $k$.
- $f_{k|k-1}(x|\xi)$ is the single-target state transition density, which models the motion of a target from state $\xi$ to state $x$.

The second term, $\gamma_k(x)$, represents the birth model, a crucial component that introduces new tracks for targets appearing for the first time. The birth intensity is often modelled as a Poisson RFS or, more commonly in practical implementations, as a Gaussian mixture where each component represents a potential birth location, state, and covariance [24]. The design of the birth model is critical; a poorly structured model can lead to a high incidence of false tracks and a long confirmation time for true ones. To mitigate this, adaptive and measurement-driven birth models have been developed. These models dynamically create new birth components from measurements that are not associated with existing tracks, thereby improving the filter's responsiveness to the appearance of new targets in unexpected locations [21].

*Update Step*
The update step is where the extended target model most significantly departs from the point target filter. It corrects the predicted intensity using the current set of measurements, $Z_k$. Since a single extended target can generate a set of measurements, the update must evaluate the likelihood of the entire measurement set, not just individual measurements.

A foundational assumption is that the number of measurements generated by a target with state $x$ follows a Poisson distribution with an average rate, $\lambda(x)$. The update equation must therefore consider all possible ways to partition the measurement set $Z_k$ into subsets, where each subset could have originated from a specific target or from clutter. This makes the update computationally demanding. For a set of measurements $Z$, the updated PHD is given by:

$$D_{k|k}(x) \propto e^{-\lambda(x)} p_{D,k}(x) D_{k|k-1}(x) \sum_{\mathcal{P} \uplus \{\mathcal{W}_1,\ldots,\mathcal{W}_{|\mathcal{P}|}\}=Z} \prod_{j=1}^{|\mathcal{P}|} \frac{\lambda(x) g_k(\mathcal{W}_j|x)}{\kappa(\mathcal{W}_j)} \tag{2.4}$$

This is a conceptual representation; the exact form varies by implementation [24].

The update can be understood as follows:

- The likelihood of a target being detected, $p_{D,k}(x)$, and generating zero measurements is accounted for, which contributes to the missed detection term (not explicitly shown in the simplified form above but inherent in the full derivation).
- The core of the update is a sum over all possible partitions, $\mathcal{P}$, of the measurement set $Z_k$.
- For each partition, each subset of measurements $\mathcal{W}_j$ is evaluated. The term $g_k(\mathcal{W}_j|x)$ is the likelihood that the subset $\mathcal{W}_j$ was generated by the target $x$. This is weighed against the likelihood that the subset is purely clutter, $\kappa(\mathcal{W}_j)$.
- This process reinforces the intensity $D_{k|k-1}(x)$ in regions of the state space where the predicted extent aligns well with received measurement subsets.

Due to the combinatorial nature of measurement partitioning, exact solutions are intractable. Practical implementations, such as the Gaussian Mixture-PHD filter, use approximations. A common strategy is to partition the predicted Gaussian components representing a target's extent and then assign measurements to these components, simplifying the likelihood calculation and making the filter feasible, albeit at the cost of increased computational load that necessitates aggressive pruning and merging strategies [24].

*Prune and Merge Step*

To manage the computational complexity and prevent an unbounded growth in the number of Gaussian components, the PHD filter incorporates a prune and merge step after the update. This step involves removing components with weights below a certain threshold and merging components that are close in the state space. This helps maintain a manageable number of components while preserving the essential structure of the intensity function.

Fundamental Limitations

The main limitation of the PHD filter arises from its core approximation: it propagates only the first-order moment (the intensity function) of the multi-target density. While this intensity, $D(x)$, provides the expected number of targets and their likely states, it does not retain information about the variance or higher-order statistics of the target count (i.e., the cardinality distribution) [23], [24].

This leads to two significant consequences:

- **Cardinality Estimation:** The PHD filter's estimate of the number of targets, obtained by integrating the intensity over the state space, $\hat{N}_k = \int D_{k|k}(x)dx$, typically exhibits higher variance compared to more advanced methods. It is sensitive to clutter and missed detections, resulting in less stable estimates of the target count across time steps.
- **Track Continuity and Identity Management:** Since the PHD filter propagates an undifferentiated density of expected targets, it does not maintain distinct track identities over time. State estimates are extracted from the peaks of the intensity at each step, but there is no explicit mechanism to associate a state estimate at time $k$ with a specific estimate from time $k-1$. As a result, the filter is "anonymous" and can suffer from track fragmentation or coalescence, particularly when targets are closely spaced or their trajectories intersect. Track stitching and identity management must be handled by post-processing heuristics, which are not part of the filter's core recursion [23], [24].

### 2.4.4. The Poisson Multi-Bernoulli Mixture Filter

A more recent and advanced approach within the RFS framework is the Poisson Multi-Bernoulli Mixture (PMBM) filter [6], [25], [26]. The PMBM filter provides an elegant and effective method for tracking a time-varying and unknown number of targets by separating the set of all targets into two distinct categories: those that have not yet been detected and those that have been detected at least once [25].

#### Core Concept

The fundamental idea of the PMBM filter is to model the multi-target state density as a convolution of two independent RFSs: a Poisson Point Process (PPP) and a Multi Bernoulli Mixture (MBM) [6], [25], [26].

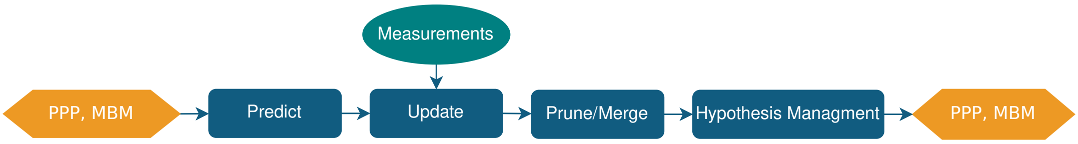


**Figure 2.8:** Schematic of the PMBM update step: measurements Z are associated to the undetected PPP and the detected MBM, yielding updated PPP intensity and updated MBM hypotheses. Note: this is a simplified illustration and does not capture all complexities of the PMBM update process.

- Undetected Targets (PPP): The set of targets that exist but have not yet been detected by the sensor is modelled as a PPP. The PPP is characterised by an intensity function, $\lambda_u(\cdot)$, which describes the expected number and spatial distribution of these potential targets. This component also naturally handles the birth of new targets, which are initially considered undetected and are modelled by a PPP with a birth intensity $\lambda_B(\cdot)$ [25], [27].
- Detected Targets (MBM): The set of targets that have been detected at least once is modelled as an MBM. An MBM is a weighted sum of multi-Bernoulli (MB) densities, where each MB density is a union of independent Bernoulli RFSs [28]. Each Bernoulli component represents a single potentially detected target and is defined by its existence probability, r, and its state probability density function, $p(x)$. Each component in the mixture corresponds to a unique global data association hypothesis, with an associated weight representing its likelihood [25].

The overall PMBM density for a set of targets $X$ is thus a convolution of the densities for the undetected set $X_u$ and the detected set $X_d$:

$$f(X) = \sum_{X_u \uplus X_d = X} f_p(X_u) f_{\text{mbm}}(X_d) \tag{2.5}$$

where $\uplus$ denotes the disjoint union[27].

#### PMBM Filter Recursion

The PMBM filter propagates the multi-target density, which is fully defined by its parameters, through a recursive prediction and update cycle. The core strength of this filter is that both the predicted and updated densities retain the PMBM form, a property known as conjugacy [6], [25].

*Prediction Step*

Given the posterior PMBM density from the previous time step, the prediction step computes the prior density for the current time step. This involves predicting the evolution of both the undetected and detected target sets according to the standard multi-target dynamic model, which includes target survival, motion, and birth [6], [28].

Undetected Targets (PPP): The predicted intensity of the PPP, $\lambda_u^+(\cdot)$, is the sum of two components: the intensity of surviving undetected targets from the previous step and the intensity of new targets born at the current time step, $\lambda_b(\cdot)$ [25], [28]. The prediction for the surviving targets is governed by the survival probability $p_S(\cdot)$ and the single-target transition density $f_{k|k-1}(\cdot|\cdot)$. The expression is:

$$\lambda_u^+(x) = \lambda_b(x) + \int p_S(x')f_{k|k-1}(x|x')\lambda_u(x')dx' \tag{2.6}$$

where the integral denotes the expected contribution from surviving undetected targets [25].

Detected Targets (MBM): Each existing Bernoulli component within the MBM is predicted forward in time. The prediction step does not change the number of global hypotheses or the number of Bernoulli components within each hypothesis [25]. For each Bernoulli component $(r_{j,i}, f_{j,i})$ in a global hypothesis $j$, the predicted existence probability $r_{j,i}^+$ and state density $f_{j,i}^+(\cdot)$ are calculated based on the probability of survival and the motion model [25], [28]. The hypothesis weights $W_j$ remain unchanged during this step [25].

*Update Step*
Given the predicted (prior) PMBM density and a new set of measurements Z, the update step computes the posterior PMBM density using Bayes' rule. This step involves considering all possible associations between measurements and targets, which leads to a significant increase in the number of hypotheses [25].

- Undetected Targets (PPP): The intensity of the PPP for undetected targets is updated to account for the possibility of missed detections. Essentially, the predicted intensity is scaled by the probability of missed detection, $q_D(x)$, which is the probability of a target generating an empty measurement set [25]. The updated intensity is:

$$\lambda_u(x) = q_D(x)\lambda_u^+(x) \tag{2.7}$$

- Detected Targets (MBM): The update for the MBM component is more complex as it must account for all data associations. For each predicted global hypothesis $j$, a new set of updated global hypotheses is created, one for each valid association $A$ of measurements in $Z$ to the targets in that hypothesis [25].
- - New Bernoulli Components: Each non-empty subset of measurements can be hypothesized to have originated from a previously undetected target. This creates new Bernoulli components, initiated from the predicted PPP [6].
  - Existing Bernoulli Components: Each existing Bernoulli component from the prediction is updated in one of two ways:
    * Missed Detection: The target was not detected. The Bernoulli parameters are updated accordingly, typically resulting in a lower existence probability [6].
    * Detection: The target is associated with a specific measurement (or a subset of measurements for extended targets). The Bernoulli component is updated using this measurement, typically resulting in an existence probability of 1 and a refined state density [6].
  - Hypothesis Weights: The weight $W_{A_j}$ for each new global hypothesis is calculated based on the likelihood of that specific data association. This likelihood is a product of the likelihoods $L_C$ for each cell $C$ in the measurement partition, which can correspond to a new target detection, an existing target detection, or a missed detection [25]. The final weights are normalized across all new global hypotheses from all predicted hypotheses.

### Advantages of the PMBM Filter

The PMBM framework offers several significant advantages over other multi-target tracking filters, such as the PHD or JPDA filters. A key theoretical advantage is that the PMBM density functions as a conjugate prior for the standard multi-target dynamic and measurement models, a property which holds for both point and extended targets [25], [27]. This conjugacy is crucial because it means that if the prior density is a PMBM, the posterior density resulting from the prediction and update steps also remains a PMBM. This stability allows for the derivation of closed-form recursive expressions, thereby providing an explicit representation of the posterior density without requiring approximations of its functional form [6].

Furthermore, the structure of the filter facilitates more efficient hypothesis management by separating targets into detected and undetected sets. The use of a Poisson Point Process for the undetected targets provides an effective mechanism to handle potential new targets without creating an excessive number of hypotheses [6]. This PMBM structure has also been shown to be more efficient than that of the $\delta$-GLMB filter, as the PMBM often requires fewer hypotheses to represent the same density [25].

Finally, regarding practical application, simulation studies have consistently shown that the PMBM filter performs well in comparison to other state-of-the-art filters. It frequently outperforms PHD, JPDA, and GNN filters, yielding smaller estimation errors. This superior performance is particularly notable in challenging scenarios characterised by heavy clutter and frequent target birth and death events [28].

#### Limitations and Practical Considerations

Despite its theoretical elegance and strong performance, the PMBM filter faces practical challenges, primarily related to computational complexity. The number of global hypotheses (i.e., components in the MBM) can grow exponentially with the number of measurements and detected targets. This is because each new set of measurements creates a multitude of new data association possibilities, leading to a rapid increase in the number of terms in the MBM density [25]. Specifically, the number of unique Bernoullis required in the exact PMBM filter equates to the number of possible subsets of all measurements gathered up to the current time step [25].

This inherent combinatorial expansion means that computational tractability is not guaranteed. Therefore, any practical implementation of the PMBM filter must necessarily use approximation techniques to manage this complexity. Common strategies often begin with gating, which discards unlikely measurement-to-target pairings based on their spatial distance or association probability, thereby reducing the number of associations to consider [29]. Following this, hypothesis pruning is frequently applied, which involves removing entire global hypotheses if their corresponding weights fall below a predetermined threshold, effectively discarding the least likely interpretations of the data [30].

To avoid enumerating all association possibilities in the first place, efficient ranking and selection algorithms, such as the well-known Murty's algorithm, are utilized to find the k-best association hypotheses directly [6], [26]. Furthermore, complexity is managed through merging techniques. This involves combining Bernoulli components, or specific components like GGIW components within the mixture, that are deemed sufficiently similar. This similarity is often quantified by minimising the Kullback-Leibler divergence (KLD) between them, which serves to reduce the total component count [25], [28]. Finally, a strategy known as recycling is often used to manage components representing targets with very low certainty; this involves approximating Bernoulli components that possess very low existence probabilities as a Poisson Point Process and adding them back into the intensity function of the undetected targets [25].

By applying these methods, the PMBM filter can achieve a better balance between tracking accuracy and computational cost, making it a state-of-the-art solution for multiple extended object tracking problems [29].

# 3 Methodology

In this chapter we describe the implementations, models, scenarios, hyperparameter optimisation, and evaluation protocol. The limits imposed on the methods and how to ensure a fair comparison are also discussed.

## 3.1. Core Tracking Frameworks

To address the research questions comprehensively, three distinct multi-object tracking paradigms were implemented. These were chosen to span a representative spectrum of approaches, from classical, explicit data association methods to modern, implicit techniques based on Random Finite Sets theory. This selection facilitates a thorough comparison of fundamentally different tracking philosophies. All three frameworks were re-implemented in python within a common simulation and data-processing environment to ensure consistency and fairness in evaluation.

### 3.1.1. The Baseline Tracking-by-Detection Approach

The first framework serves as a baseline and is implemented as a traditional Tracking-by-Detection (TBD) pipeline. This paradigm explicitly separates the tasks of detection, data association, and state estimation. In each frame, a standard DBSCAN algorithm is utilised to cluster the raw 4D radar point cloud into a set of discrete object detections. Subsequently, the measurement-to-track association problem is resolved using the Hungarian Algorithm, a robust and classical solution for assignment problems in cluttered environments. The cost function for this assignment is based on the Euclidean distance between predicted and measured states. Finally, the state of each individual track is maintained and propagated over time using either a standard Kalman Filter for linear models or an Unscented Kalman Filter for non-linear models, ensuring appropriate state estimation for the chosen motion and extent models.

### 3.1.2. The Probability Hypothesis Density Filter

The second framework is the Probability Hypothesis Density (PHD) filter, a RFS-based method that bypasses explicit data association. Instead of managing individual tracks, the PHD filter propagates the first-order statistical moment, known as the intensity function, of the multi-target posterior distribution. This function represents the density of expected objects across the state space. By modelling both true targets and clutter within a unified probabilistic framework, the PHD filter offers an elegant and computationally efficient alternative to traditional TBD. The specific implementation used in this work is a Gaussian Mixture PHD (GM-PHD) filter, where the intensity function is approximated by a mixture of Gaussian components. This approach is based on the formulation presented by Granström et al. [13], with the full algorithmic details provided in Appendix A.

The performance of the GM-PHD filter is dependent on some implementation choices, particularly the target birth model and the measurement clustering method. The birth model is responsible for introducing new Gaussian components into the state during the prediction step, representing potential new targets. The design of this model involves a trade-off: an overly dense birth model can create excessive components and increase computational load, while a sparse model may fail to initiate tracks for newly appearing targets correctly, thereby degrading performance. A simple model placing birth components only at the edges of the sensor's field of view is suboptimal in this case, as targets can already be present within the field of view before they are first detected. Therefore, this work employs an angular-sector grid model, which partitions the entire field of view into 16 sectors, as illustrated in Figure 3.1. Each sector defines a birth component with specific spatial uncertainty, ensuring that new tracks can be initiated anywhere in the sensor's range.

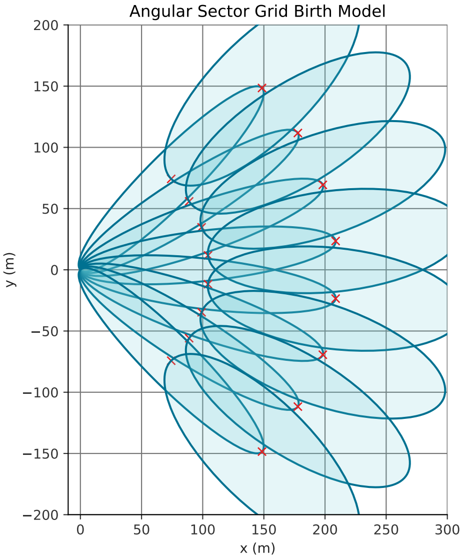


**Figure 3.1:** Angular-sector birth model used for the GM-PHD filter. The field of view is partitioned into 16 angular sectors; each sector defines a birth component with associated spatial priors. Elipses are drawn at one standard deviation.

The second choice is the measurement clustering technique. To reduce the combinatorial complexity of associating every measurement with every Gaussian component, measurements are first grouped into clusters, where each cluster likely originates from a single source. This work adopts a multi-scale DBSCAN approach, a method also utilised in [13] and [27]. By running the DBSCAN algorithm multiple times with different distance parameters, several alternative partitions of the measurement set are generated. This provides a robust way to handle variations in measurement density and object extent. While other clustering techniques exist, this method was selected for its proven effectiveness and its applicability across different RFS filters, including the PMBM filter discussed later. This consistency in pre-processing ensures a more direct and fair comparison of the core filtering algorithms.

### 3.1.3. The Poisson Multi-Bernoulli Mixture Filter

The third and most sophisticated framework implemented is the Poisson Multi-Bernoulli Mixture (PMBM) filter, a state-of-the-art RFS-based tracker. The PMBM filter provides a more refined approach than the PHD filter by partitioning the set of objects into two distinct populations. Previously undetected objects, for which there is no prior information, are modelled as a Poisson Point Process. Conversely, objects that have been detected at least once are modelled as a set of individual Bernoulli RFS, each with its own probability of existence and state distribution. This partitioned structure allows for more explicit track management and data association hypotheses within the RFS framework, providing a richer representation of the multi-object state. The implementation for this work follows the work of Xia et al. [31], Granström et al. [25], and Garcia-Fernandez et al. [6], with a detailed description available in Appendix B.

For the implementation of the PMBM filter the same clustering technique and birth model are used as described in subsection 3.1.2. This ensures that the pre-processing steps are consistent across both RFS-based methods, allowing for a more direct comparison of their core filtering algorithms.

## 3.2. State and Extent Representation Models

In this thesis we test how state and extent model choice affects tracking accuracy in maritime data. To investigate this, each of the three core tracking frameworks was paired with a variety of motion and extent models, ranging from simple to complex. This combinatorial approach allows for a systematic analysis of how model choice interacts with the tracking algorithm's performance. The full derivations of all motion models and their corresponding process noise models are presented in Appendix F.

### 3.2.1. Extended Object Models

Three distinct models were implemented to represent the physical extent of maritime vessels.

- **Random Matrix or Inverse Wishart (GIW) Model:** A random matrix-based using an Inverse Wishart Distribution to probabilistically represents an object's elliptical shape. Its key advantage is that it forms a conjugate prior within the Bayesian framework, allowing for efficient and closed-form linear updates. The implementation details are given in Appendix C.
- **Gaussian Process (SCGP) Model:** A flexible, non-parametric model used to learn and represent arbitrary, non-elliptical object shapes directly from the sensor's point cloud measurements. This model is capable of capturing more complex vessel geometries. The specific implementation is detailed in Appendix D. Used in this work is the EKF implementation of the extent model. The UKF version was implemented and tested but wasn't feasible to use in practice due to computational constraints. For a similar reason a symmetric basis function is used.
- **Principal Axis (PA) Model:** A simple parametric model that approximates an object's extent as an ellipse, estimating its centre, semi-axis lengths, and orientation. This serves as a baseline for extent estimation.

### 3.2.2. Motion Models

Four different motion models were employed to capture the kinematic behaviour of vessels. These models are categorised by their assumption of linearity.

- **Linear Models:** These models are suitable for objects with simple dynamics.
  - **Constant Velocity (CV):** Assumes an object moves in a straight line at a constant speed. It is the simplest model and serves as a fundamental baseline.
  - **Constant Acceleration (CA):** Extends the CV model to account for linear acceleration, making it suitable for objects that are speeding up or slowing down.
- **Non-Linear Models:** These are required to accurately model more complex vessel manoeuvres.
  - **Constant Turn Rate and Velocity (CTRV):** Models objects moving along a circular path with a constant turn rate and speed, which is characteristic of coordinated turns.
  - **Constant Turn Rate and Acceleration (CTRA):** The most complex model evaluated, allowing for simultaneous changes in both speed and turn rate. This model is designed to capture highly dynamic and non-linear manoeuvres.

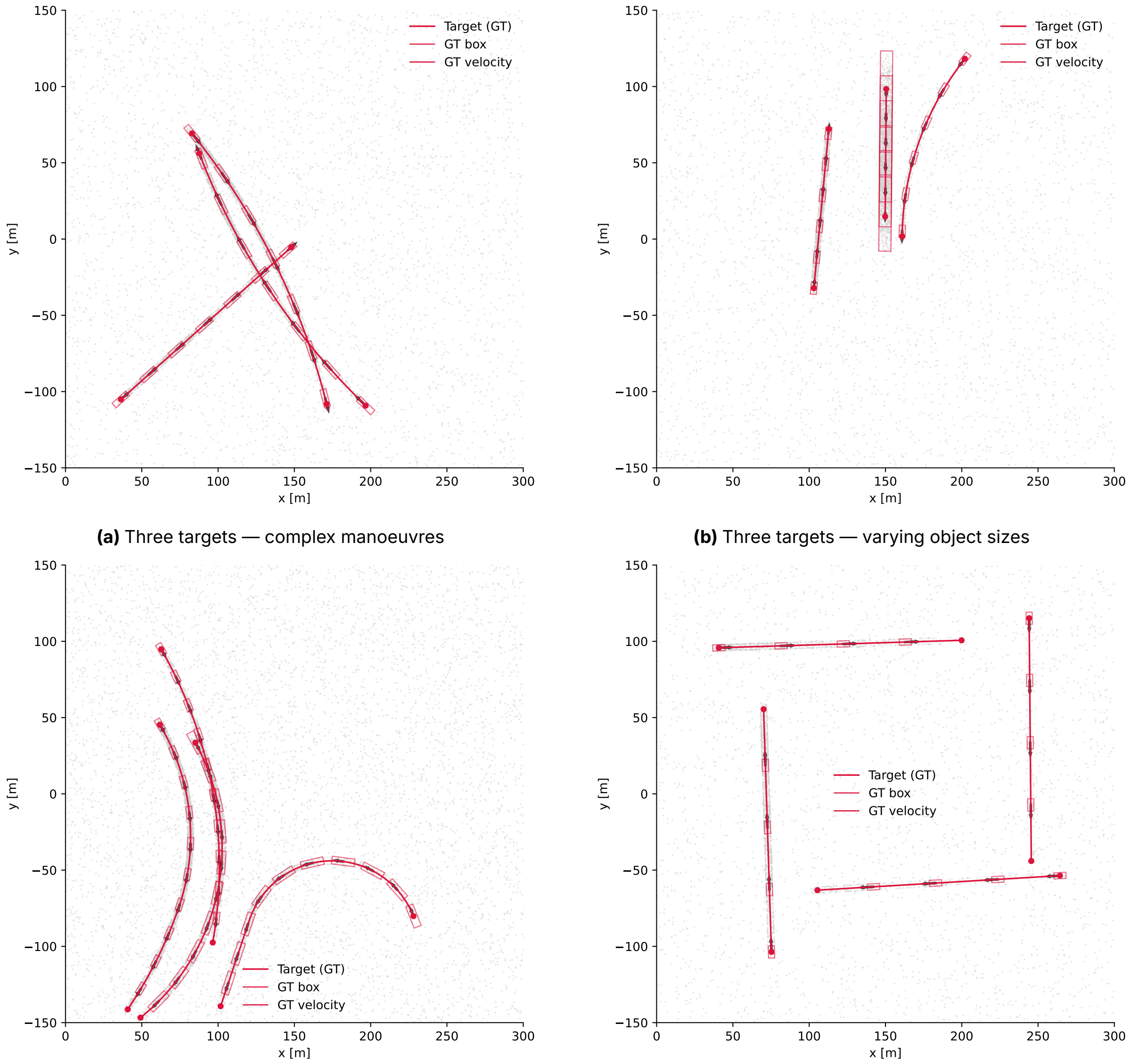
## 3.3. Experimental Design and Data

The evaluation was conducted in two distinct phases to ensure both a controlled, systematic comparison and a robust validation against real-world complexities. The first phase utilised simulated data for rigorous quantitative analysis, while the second phase used real-world data to verify the applicability of the findings.

### 3.3.1. Simulated Data

A custom simulator was developed to generate realistic 4D millimeter wave radar point clouds, though not physically correct, for a series of predefined maritime scenarios. This simulation environment provides perfect ground truth for all object states, including their kinematics and extent, as well as the associated point cloud measurements. This enables precise and unambiguous error calculation. The simulator accounts for key radar characteristics, including discrete range and velocity bins, signal fall-off with distance, clutter noise, and angular noise, though it does not model angular binning.

The simulated scenarios were designed to encompass a range of critical and challenging maritime situations, such as head-on approaches, overtaking manoeuvres, crossing paths, and interactions near static infrastructure like buoys and docks. These scenarios, exemplified in Figure 3.2, were crafted to stress-test the tracking frameworks under varied conditions of target count, motion complexity, and object extent.



**(a)** Three targets — complex manoeuvres

**(b)** Three targets — varying object sizes

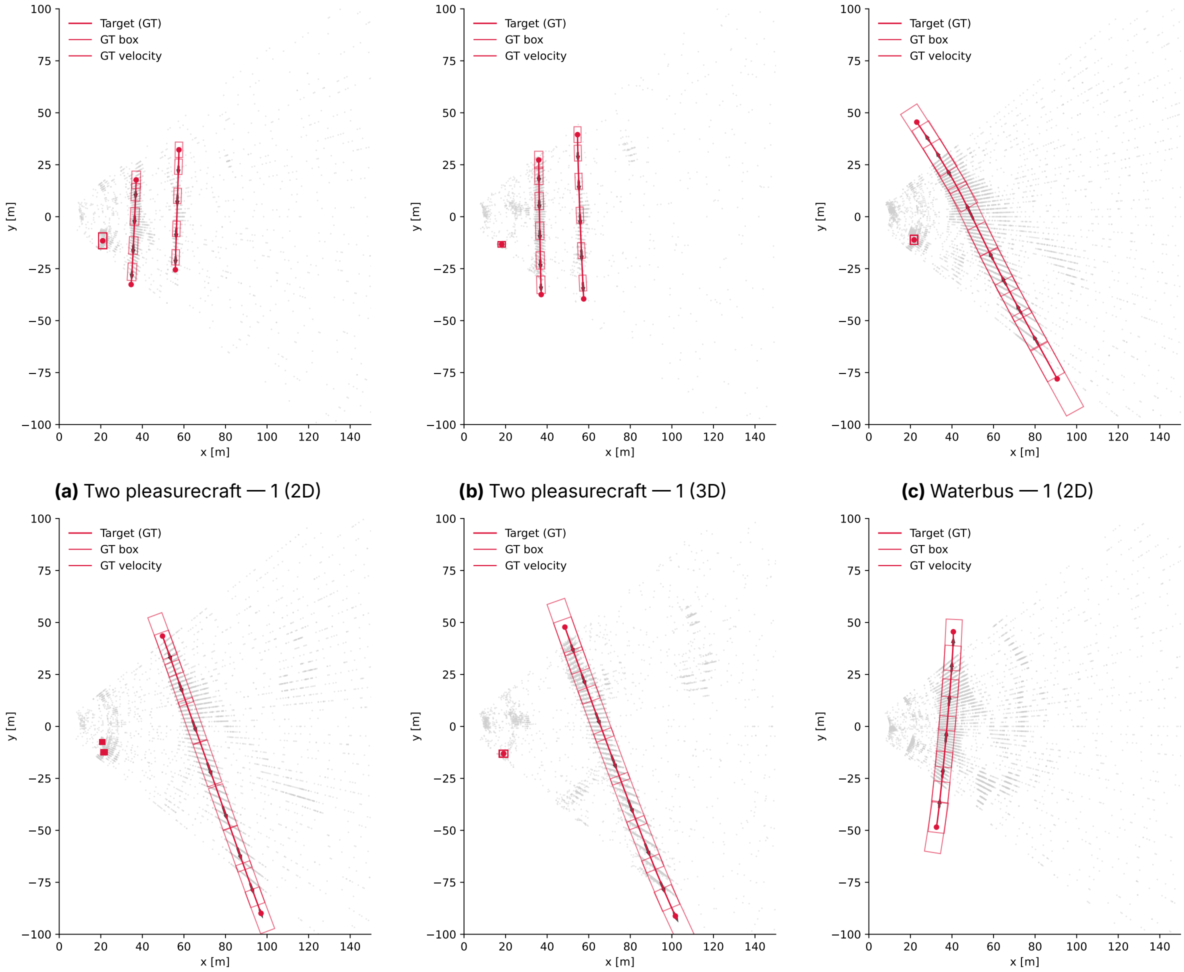
**(c)** Four targets — complex manoeuvres

**(d)** Four targets — rectangle formation, CV motion

**Figure 3.2:** Example simulated tracking scenes used in the evaluation. Each panel shows ground truth for the indicated scenario. These figures illustrate varied target counts, motion complexity, and object extents used to stress-test the tracking frameworks.

### 3.3.2. Real-World Data

For the second phase of the evaluation, a dataset was collected using a 4D millimeter wave radar mounted on a vessel operating in an inland waterway. This phase serves as a crucial validation step, intended to confirm that the conclusions drawn from the simulated experiments are robust and applicable under real-world conditions. This dataset features genuine sensor noise, environmental clutter, and the unpredictable behaviour of other vessels, providing a final test of the algorithms' practical viability. Example scenes from this dataset are shown in Figure 3.3.



**(a)** Two pleasurecraft — 1 (2D)

**(b)** Two pleasurecraft — 1 (3D)

**(c)** Waterbus — 1 (2D)

**(d)** Waterbus — 2 (2D)

**(e)** Waterbus — 2 (3D)

**(f)** Waterbus — 3 (2D)

**Figure 3.3:** Selected experimental tracking scenes from the real-world dataset. Each panel shows ground truth and estimated tracks for the indicated encounter (2D views and 3D visualisations where available).

## 3.4. Hyperparameter Optimisation for Fair Comparison

A critical component of this methodology is ensuring that every algorithm is evaluated at its optimal performance. A naive comparison using default or arbitrarily chosen parameters would be scientifically unsound and could lead to misleading conclusions. To address this, a rigorous hyperparameter optimisation process was conducted for every unique combination of tracking framework, motion model, and extent model.

This process was automated using the Optuna framework[32]. For each specific combination, a separate optimisation study was performed on the simulated dataset. Optuna systematically explores the high-dimensional parameter space of the algorithm—including parameters such as process noise, detection thresholds, and gating parameters—with the objective of finding the hyperparameter set that minimises the GOSPA tracking error, as defined in section 3.5.

Furthermore, to gain insight into the tuning process, parameter importance was analysed using PED-ANOVA (Pearson Divergence ANOVA). This technique identifies which hyperparameters are most critical for achieving top-tier performance by comparing the distributions of parameters from high-performing trials against the distributions from all trials. This allows for the identification of parameters that require precise tuning. This meticulous process yields two key outputs: first, the best possible hyperparameter configuration for each algorithm combination, ensuring a fair and robust comparison; and second, a valuable analysis of parameter importance, revealing which tuning parameters most significantly influence tracking performance.

## 3.5. Evaluation Protocol

To ensure consistency and objectivity across all experiments, the performance of each tracker configuration was quantified using a single, comprehensive metric.

### 3.5.1. Core Metric: GOSPA

The chosen metric is the Generalized Optimal Sub-Pattern Assignment (GOSPA)[33]. The rationale for this choice is that GOSPA provides a mathematically principled and unified framework for measuring the distance between the set of ground truth objects and the set of estimated tracks. It elegantly decomposes the total tracking error into three intuitive and interpretable components: a localisation error for correctly detected and assigned tracks, a penalty for missed objects, and a penalty for false tracks. This decomposition allows for a nuanced analysis of tracker behaviour beyond a single error value.

Once the GOSPA metric has established the optimal assignment between estimates and ground truth targets, the errors in position, velocity, orientation, and shape are quantified for the matched pairs. A crucial aspect of this evaluation protocol is that the assignment produced by the GOSPA calculation is used as the single source of truth for matching across all subsequent analyses. This ensures that when comparing different algorithms, the association between estimates and ground truth is determined by the exact same logic, thereby eliminating the assignment process itself as a confounding variable and guaranteeing a fair comparison.

*Component-wise cost used for GOSPA assignment*

Let a kinematic-extent state be $\mathbf{s} = [x,\ y,\ v_x,\ v_y,\ l,\ w,\ \psi]^\top$, with position $\mathbf{p} = (x, y)^\top$, velocity $\mathbf{v} = (v_x, v_y)^\top$, shape $\mathbf{e} = (l, w)^\top$, and orientation $\psi \in [-\pi, \pi)$. Given two states $\mathbf{x}$ and $\mathbf{y}$, the component-wise discrepancies are

$$\begin{aligned} d_{\text{pos}}(\mathbf{x}, \mathbf{y}) &= \|\mathbf{p}_x - \mathbf{p}_y\|_2, \\ d_{\text{vel}}(\mathbf{x}, \mathbf{y}) &= \|\mathbf{v}_x - \mathbf{v}_y\|_2, \\ d_{\text{shape}}(\mathbf{x}, \mathbf{y}) &= \|\mathbf{e}_x - \mathbf{e}_y\|_2, \\ d_{\text{ori}}(\mathbf{x}, \mathbf{y}) &= \big|\mathrm{wrap}(\psi_x - \psi_y)\big|, \end{aligned}$$

where the shortest signed angular difference is computed by

$$\mathrm{wrap}(\Delta\psi) = \mathrm{atan2}\big(\sin\Delta\psi,\ \cos\Delta\psi\big) \in [-\pi, \pi).$$

Each component is normalized by fixed error margins

$$m_{\text{pos}} = 5\ \mathrm{m}, \quad m_{\text{vel}} = 0.5\ \mathrm{m/s}, \quad m_{\text{shape}} = 2\ \mathrm{m}, \quad m_{\text{ori}} = 6^\circ = \frac{6\pi}{180}\ \mathrm{rad}.$$

Let $\tilde{d}_k = d_k / m_k$ for $k \in \{\text{pos}, \text{vel}, \text{shape}, \text{ori}\}$. The aggregated cost used within the GOSPA assignment is the quadratic sum

$$c(\mathbf{x}, \mathbf{y}) \ = \ \tilde{d}^2_{\text{pos}} \ + \ \tilde{d}^2_{\text{vel}} \ + \ \tilde{d}^2_{\text{shape}} \ + \ \tilde{d}^2_{\text{ori}}.$$

An error equal to its margin contributes unity to the sum, yielding balanced weighting across components.

# 4 Results

This chapter presents the quantitative and qualitative results of the experiments described in Chapter 3. Its purpose is to evaluate and compare the performance of three multiple-target tracking frameworks, Tracking-by-Detection, Probability Hypothesis Density and Poisson Multi-Bernoulli Mixture, when paired with a variety of motion and extent models and exposed to differing sensor and clutter conditions.

For reproducibility and fairness, all methods were implemented within a common simulation and data-processing framework. Hyperparameters for each approach were selected using the optimisation study reported at the start of this chapter so that comparisons reflect representative and well-tuned configurations rather than arbitrary parameter choices.

The principal quantitative metric used is the Generalized Optimal Sub-Pattern Assignment error. GOSPA provides a single, interpretable score by penalising localisation errors, missed targets and false tracks, and thus facilitates a detailed decomposition of tracking performance.

The chapter is organised as follows. It begins with the hyperparameter optimisation study that establishes a fair basis for comparison. This is followed by the core quantitative results from the simulated scenarios and a qualitative analysis of typical tracker behaviour and failure modes. Finally, the chapter concludes with an evaluation of the algorithms on the real-world dataset.

## 4.1. Parameter Study

To enable a controlled comparison, each configuration was tuned before evaluation. A direct comparison using arbitrary or default parameters would be scientifically unsound. To address this, a rigorous, automated hyperparameter optimisation process was conducted for every unique combination of tracking framework, motion model, and extent model. This study was performed exclusively on the simulated dataset, using the Optuna framework[32] to systematically search for the parameter set that minimised the mean GOSPA error accrross several scenes.

To gain deeper insight into the tuning process, the relative importance of each hyperparameter was quantified using Pearson Divergence ANOVA (PED-ANOVA). This analysis identifies which parameters are most critical for achieving high performance. The following sections present the key findings from this parameter importance study for each of the three core tracking frameworks.

For brevity and clarity the relevant graphs can be found in the appendix Appendix G. A sample graph is shown in Figure 4.1 to illustrate the format and type of information presented. Each bar represents a hyperparameter, with its width indicating the relative importance of that parameter in influencing the tracker's performance as calculated by the ANOVA. The parameters are sorted in descending order of importance, allowing for quick identification of the most critical factors affecting the tracker's accuracy and robustness.

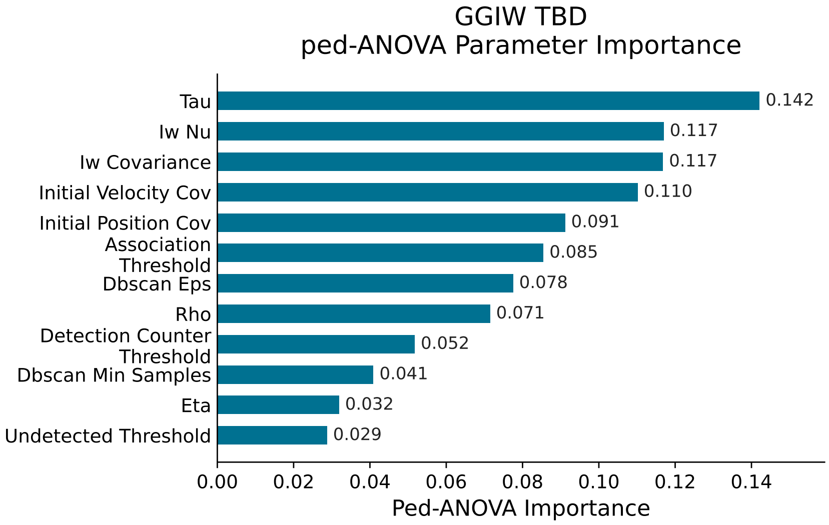


**Figure 4.1:** Random Matrix (GIW) extent model

### 4.1.1. Tracking-by-Detection

The performance of the TBD framework is highly dependent on the quality of its sequential processing stages: detection, association, and filtering. As shown in Figure G.1, the parameter importance analysis reflects the critical nature of these distinct steps.

For the TBD tracker paired with the RM extent model (Figure G.1a), the most influential parameters were Tau, IW Nu, and IW Covariance. These results indicate sensitivity to the initial extent and measurement-rate assumptions of the Random Matrix extent model. What is remarkable is the rather low importance of the undetected threshold for this filter as compared to the other methods. For the SCGP model (Figure G.1b), DBSCAN Min Samples and the Undetected Threshold dominated, indicating that tracker performance depends primarily on the quality of the initial DBSCAN point-cloud clustering.

### 4.1.2. Probability Hypothesis Density

For the PHD filter, which integrates detection and tracking, the parameter study revealed a high sensitivity to the models governing target birth and sensor measurements. As seen in Figure 4.2a for the RM PHD filter, the most critical parameters were Beta Fa, Eta, and Rho. These hyperparameters control the intensity of new "birth" components and the expected measurement rate from targets. This indicates that the PHD filter's performance is critically dependent on how it is configured to handle the appearance of new objects and how it models the radar's measurement generation process. A similar trend was observed for the SCGP PHD filter (Figure G.2b), where the birth parameter Beta Fa and the GP's own noise parameter Sigma R were dominant.

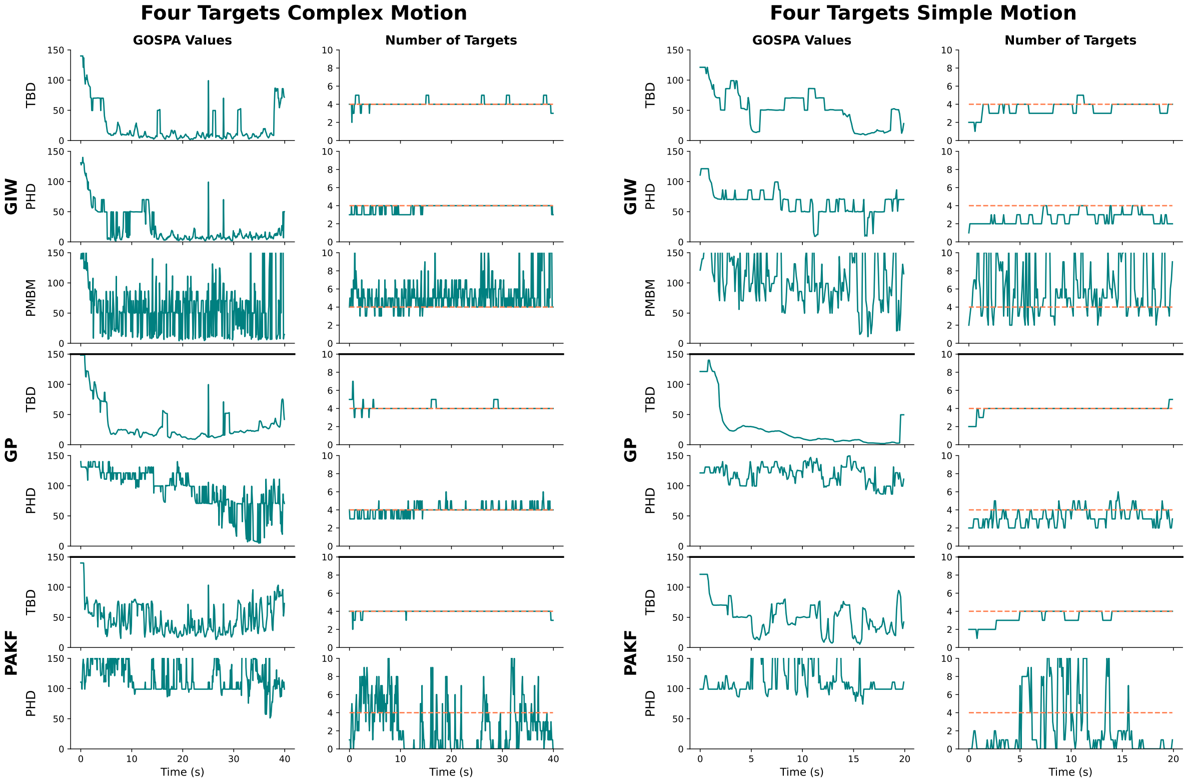
### 4.1.3. Poisson Multi-Bernoulli Mixture

Only the Random Matrix (GIW) was optimized for the PMBM use case. The other methods could not be effectively tuned due to long runtimes requiring a severely reduced range to even run. The PMBM filter represents the most complex framework, and its parameter study was confined to the RM extent model due to the prohibitive computational cost of the other configurations. The results, summarised in Figure G.3, show that the probability of detection (Pd) was overwhelmingly the most influential parameter. This suggests that the PMBM filter's performance is most sensitive to its internal model of the sensor's reliability. The parameters of the Inverse-Wishart distribution (Iw Nu, Iw Covariance), were also highly significant.

When applied to the real dataset, the optimiser consistently converged on parameter sets that resulted in zero detections. In the presence of significant real-world clutter, the GOSPA penalty for generating numerous false tracks was often higher than the penalty for simply missing all ground truth targets. Consequently, the optimiser found a local minimum by becoming highly conservative. This practical challenge is explored further in Section 5.4.1.

## 4.2. Combined Error Results

The tracking performance across various simulated and real-world scenarios is presented using the GOSPA metric. The results are grouped by scenario type.



**(a)** Four targets — complex motion

**(b)** Four targets — simple motion

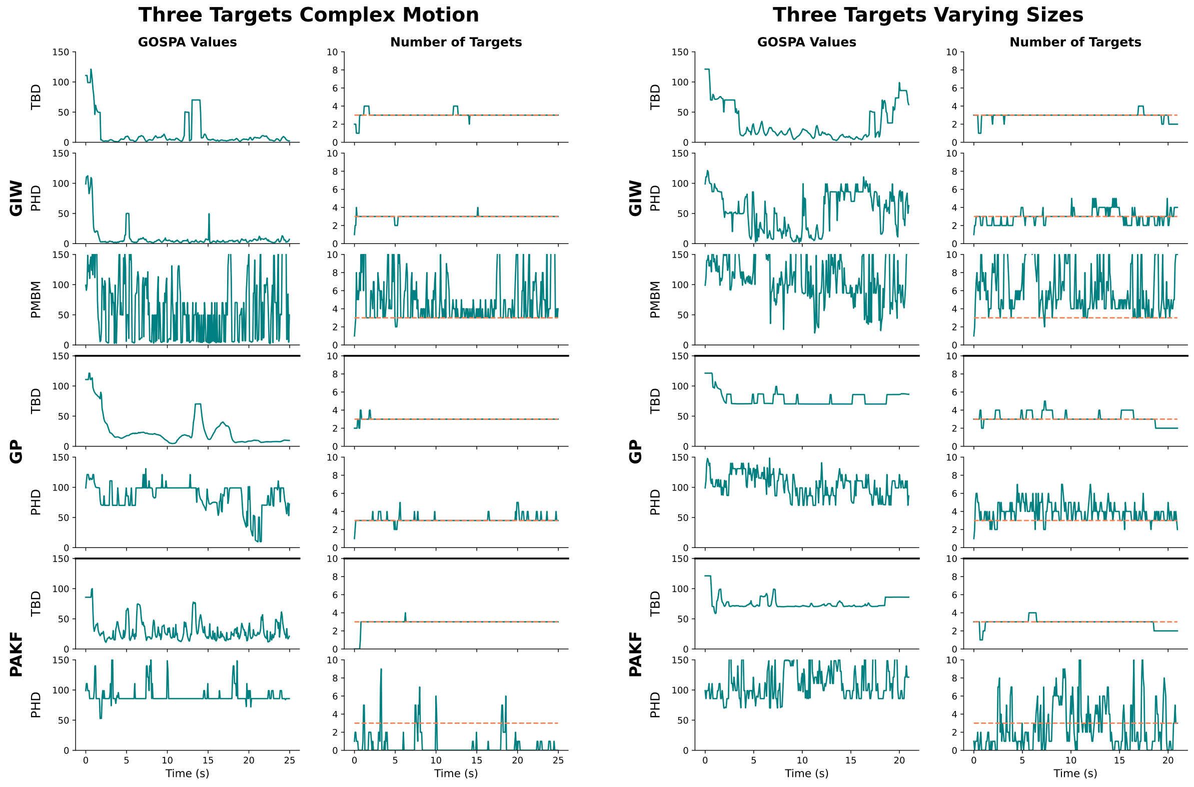
**Figure 4.2:** GOSPA combined results for the four-target simulated scenarios.

### 4.2.1. Simulated Data Observations

The results from the simulated scenarios are shown in Figure 4.2 and Figure 4.3. A consistent observation across these tests is the superior performance of the Random Matrix Model (GIW) compared to the Gaussian Process (GP) and Principal Axis (PA) models. This is evidenced by lower GOSPA scores and more stable cardinality estimates. The Tracking-by-Detection frameworks also appear to be generally well-suited for these problems.

In the four-target scenarios shown in Figure 4.2, the PMBM filter exhibits highly variable GOSPA values. However, its cardinality estimate periodically stabilises on the correct number of targets, at which points it demonstrates high accuracy. Comparing the simple and complex motion scenarios (Figure 4.2b and 4.2a respectively) reveals that the SCGP TBD filter performs better with simpler dynamics, whereas the GIW TBD filter's performance appears to degrade.

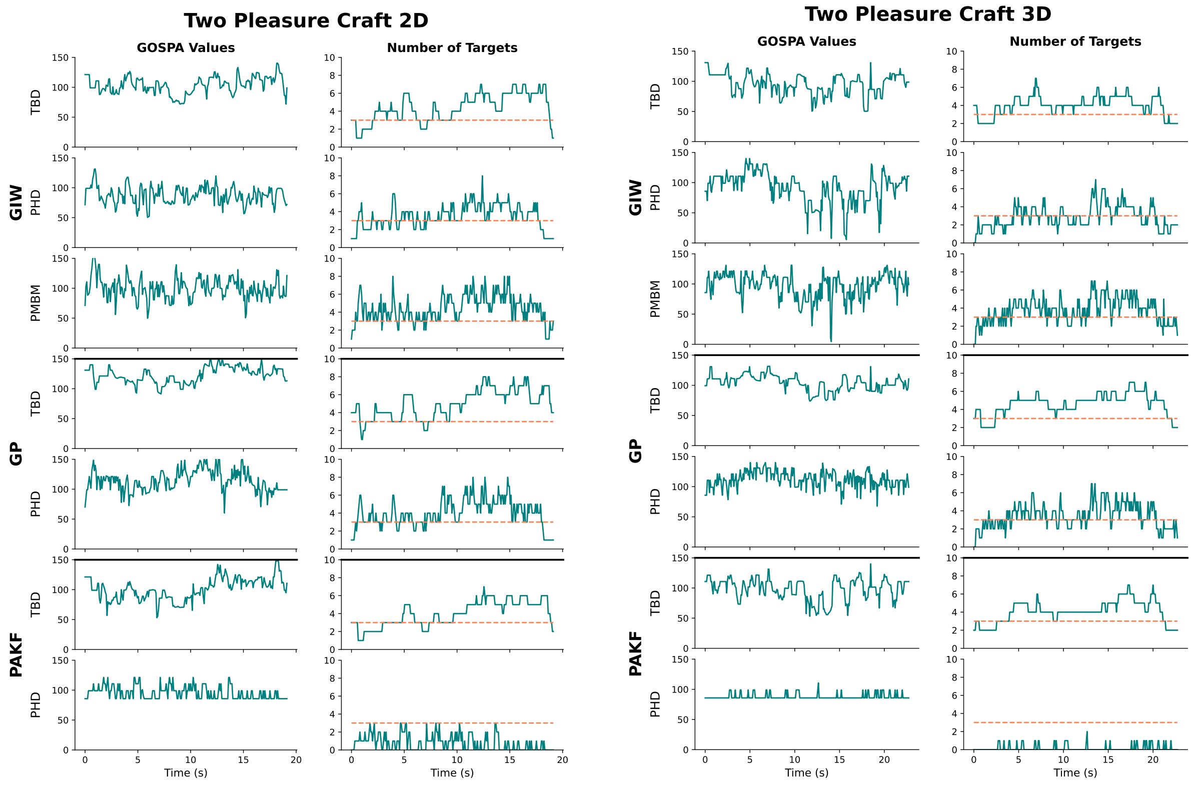
For the three-target scenarios in Figure 4.3, it is notable that the introduction of varying target sizes (Figure 4.3b) significantly worsens the performance of the TBD filters compared to the complex motion scenario with uniformly sized targets (Figure 4.3a). This suggests a challenge in tracking objects of different extents simultaneously. In the complex motion test, the PMBM filter, despite its noise, achieves moments of very low GOSPA error, indicating high potential accuracy.



**(a)** Three targets — complex motion

**(b)** Three targets — varying sizes

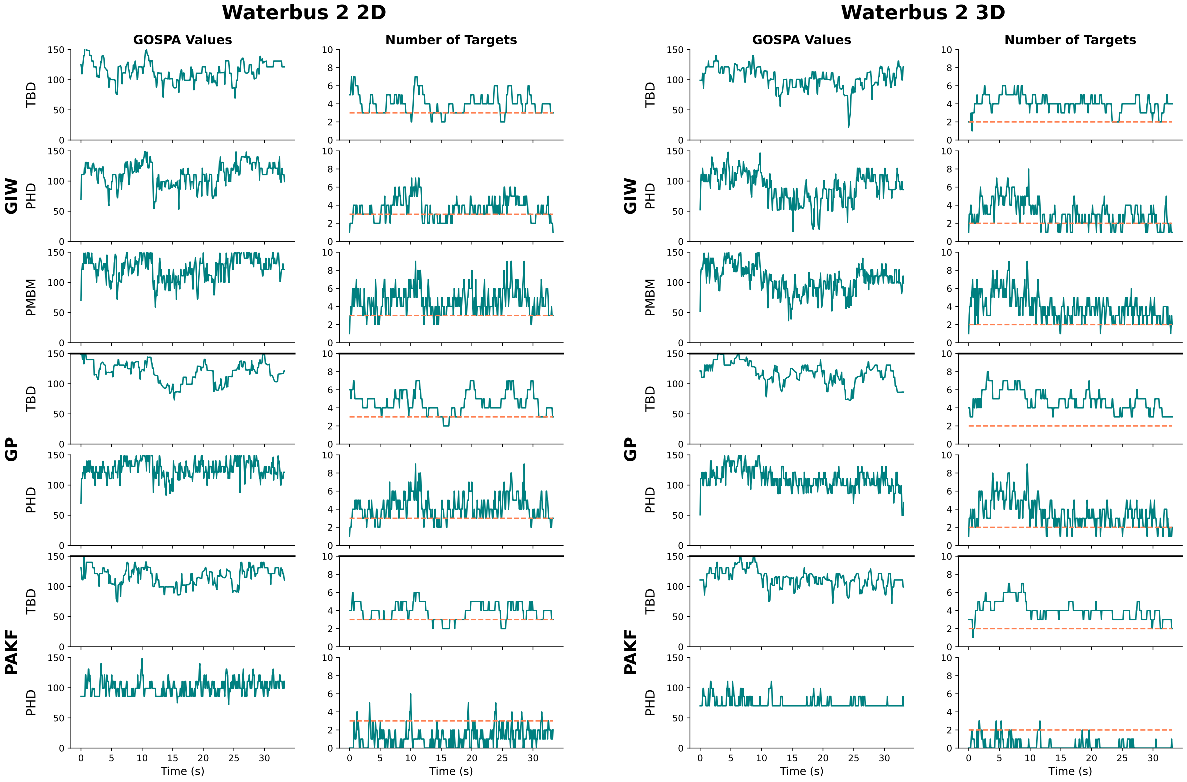
**Figure 4.3:** GOSPA combined results for the three-target simulated scenarios.



**(a)** Two pleasure craft — 2D

**(b)** Two pleasure craft — 3D

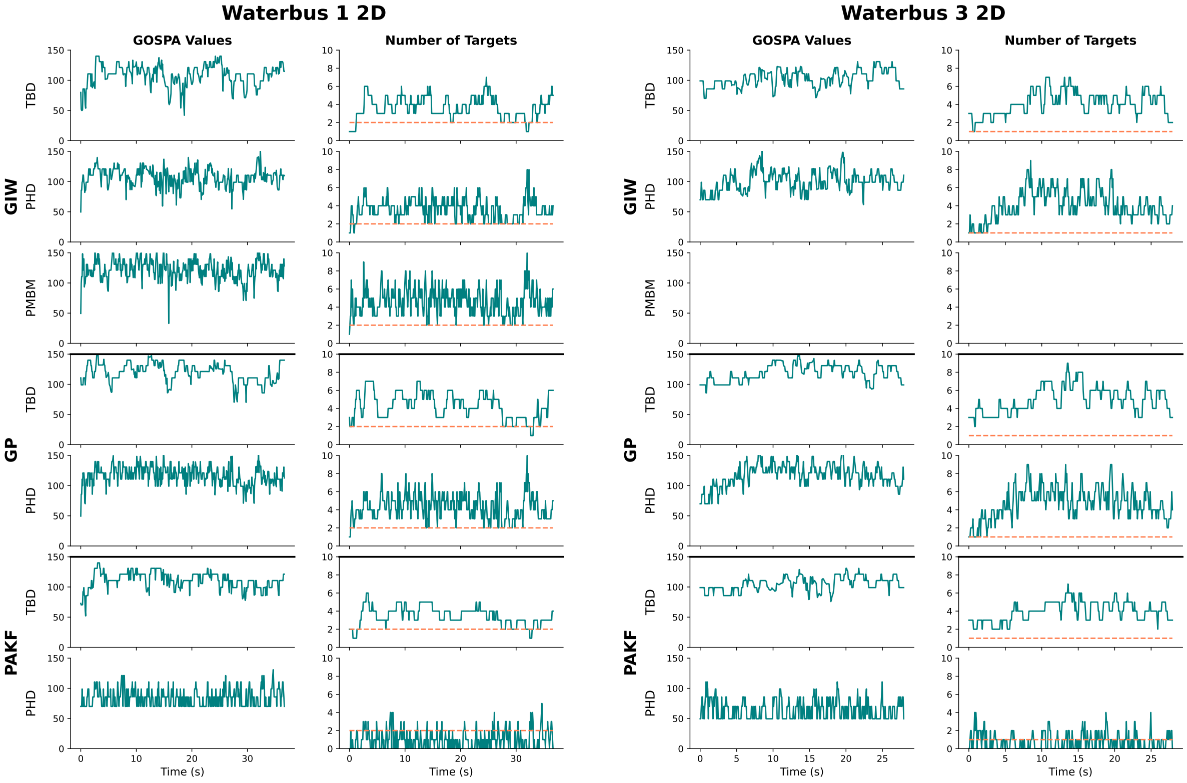
**Figure 4.4:** GOSPA combined results for the pleasure craft scenarios (2D vs 3D).



**(a)** Waterbus scenario 2 — 2D

**(b)** Waterbus scenario 2 — 3D

**Figure 4.5:** GOSPA combined results for Waterbus scenario 2, comparing 2D and 3D data.



**(a)** Waterbus scenario 1 — 2D

**(b)** Waterbus scenario 3 — 2D

**Figure 4.6:** GOSPA combined results for Waterbus scenarios 1 and 3 (2D).

### 4.2.2. Real Experimental Data Observations

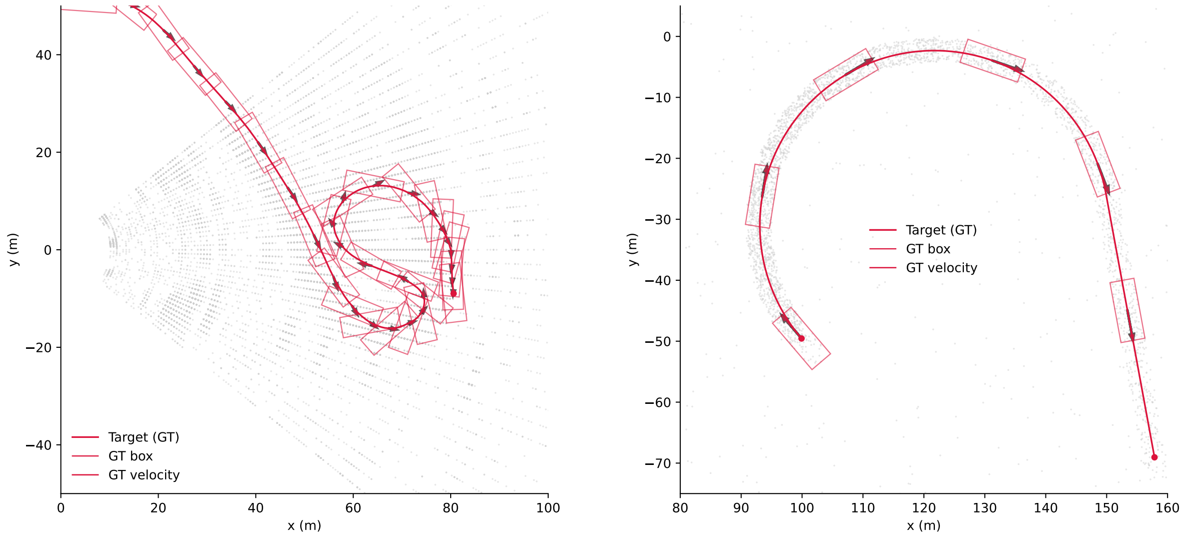
The results from the real-world vessel tracking experiments are shown in Figure 4.4. A key difference from the simulated data is the absence of a sharp initial drop in the GOSPA score. In simulation, this drop signifies the filter converging as targets appear. Its absence here may indicate that targets are present from the beginning of the sequence. The persistently high GOSPA values could also be linked to a general overestimation of the target count across most filters.

A particularly noteworthy trend is the behaviour of the PA PHD filter. Across all real data scenarios, it consistently estimates a very low, often zero, number of targets. This suggests that its parameter optimisation may have converged to a local minimum that favours non-detection to minimise error.

Finally, a direct comparison between 2D and 3D data for the 'Waterbus 2' scenario (Figures 4.5a and 4.5b) shows that the GOSPA values are generally lower for the 3D data. This performance improvement is likely due to the additional information in the 3D point clouds, which helps to reduce measurement noise and subsequently leads to fewer false tracks being initiated by the filters.

## 4.3. Motion Model Performance

To evaluate the impact of the underlying motion model on tracking accuracy, a comparative analysis was conducted. This section details the performance of four distinct motion models: two linear models, Constant Velocity and Constant Acceleration, and two non-linear models, Constant Turn Rate and Velocity and Constant Turn Rate and Acceleration. The non-linear models were implemented using both an Extended Kalman Filter and an Unscented Kalman Filter. The performance was assessed against two distinct trajectories: a simulated, predictable circular path with a switch to constant velocity and a more complex, unpredictable trajectory from a vessel recorded with a high quality on-board GPS.



**(a)** Real-data trajectory

**(b)** Simulated circular trajectory

**Figure 4.7:** The trajectories to be tracked with varying motion models.

### 4.3.1. Simulated Trajectory Analysis

The tracking errors for the simulated scenario, depicted in Figure 4.7b, are presented in Figure 4.8. The results yield a noteworthy and somewhat counter-intuitive outcome. Despite the simulation featuring a constant-rate turn followed by a straight, constant-velocity segment—dynamics for which the CTRV model is theoretically ideal—it is the linear Constant Acceleration (CA) model that demonstrates consistently superior performance.

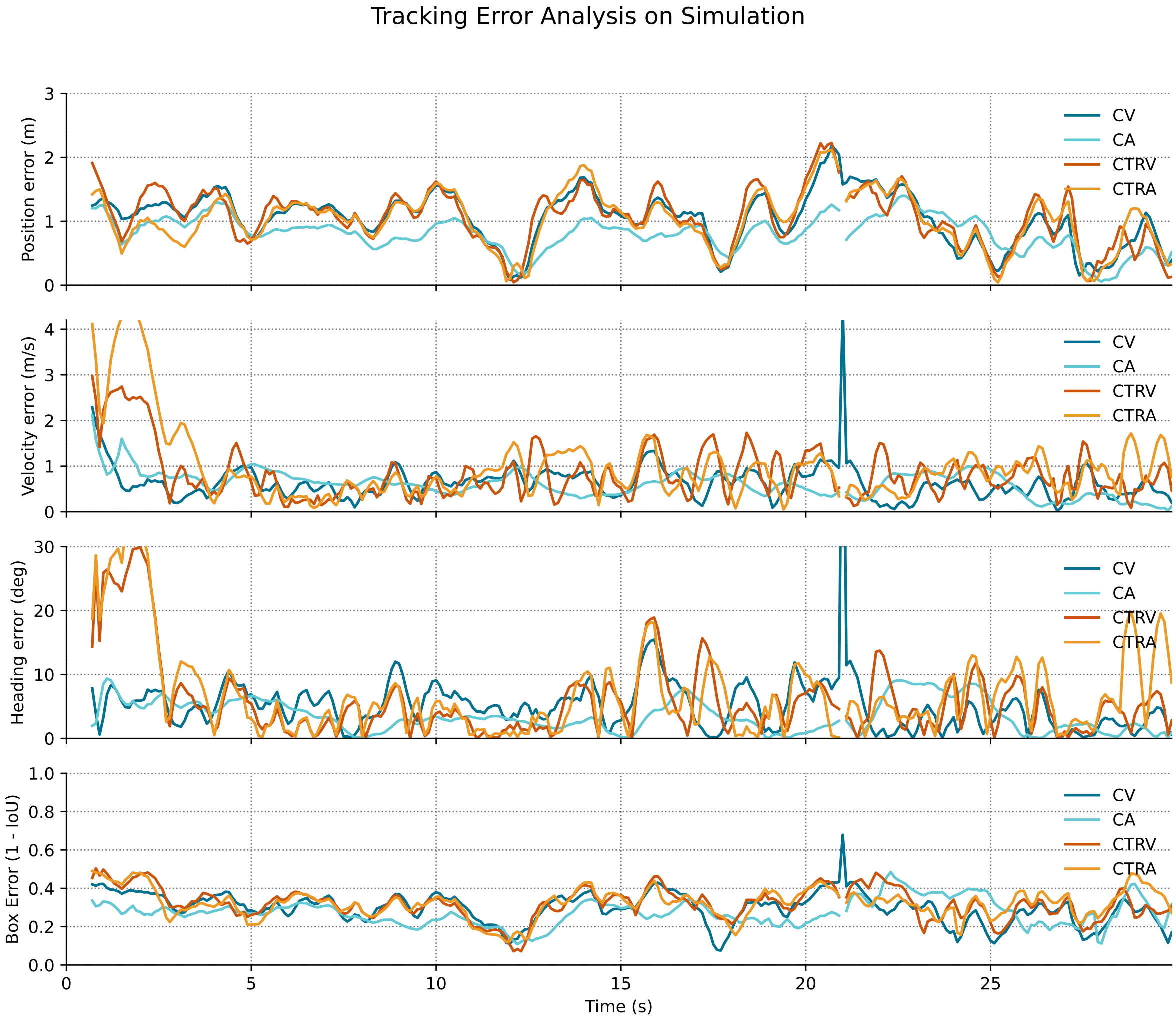


**Figure 4.8:** Tracking error comparison on the simulated circular trajectory. The subplots show positional, heading, velocity and bounding-box error over time for the indicated motion model. Note the spike in errors at 21s is due to a mistake in the groundtruth data, not the algorithms.

This outperformance is not limited to a single metric. The CA model achieves not only higher positional accuracy but also more precise estimates for both heading and velocity when compared to all other models, including the non-linear variants. This is a significant finding, as it highlights the robustness of a relatively simple linear model even in a scenario with curvilinear motion.

When comparing the two linear models, the CA model also slightly outperforms the Constant Velocity (CV) model, particularly in positional accuracy during the later stages of the track.

However, the non-linear models exhibit a distinct initialisation problem. As seen in Figure 4.8, both models display significant initial errors in heading and velocity at the moment a track is initiated and for quite a few seconds afterwards. This behaviour can be partly attributed to the initialisation of higher-order kinematic states (i.e., acceleration) at zero, which requires the filter several time steps to converge to an accurate estimate. Nevertheless, once this convergence is achieved and the track becomes more stable, though not as stable as the linear models.

### 4.3.2. Real-World Trajectory Analysis

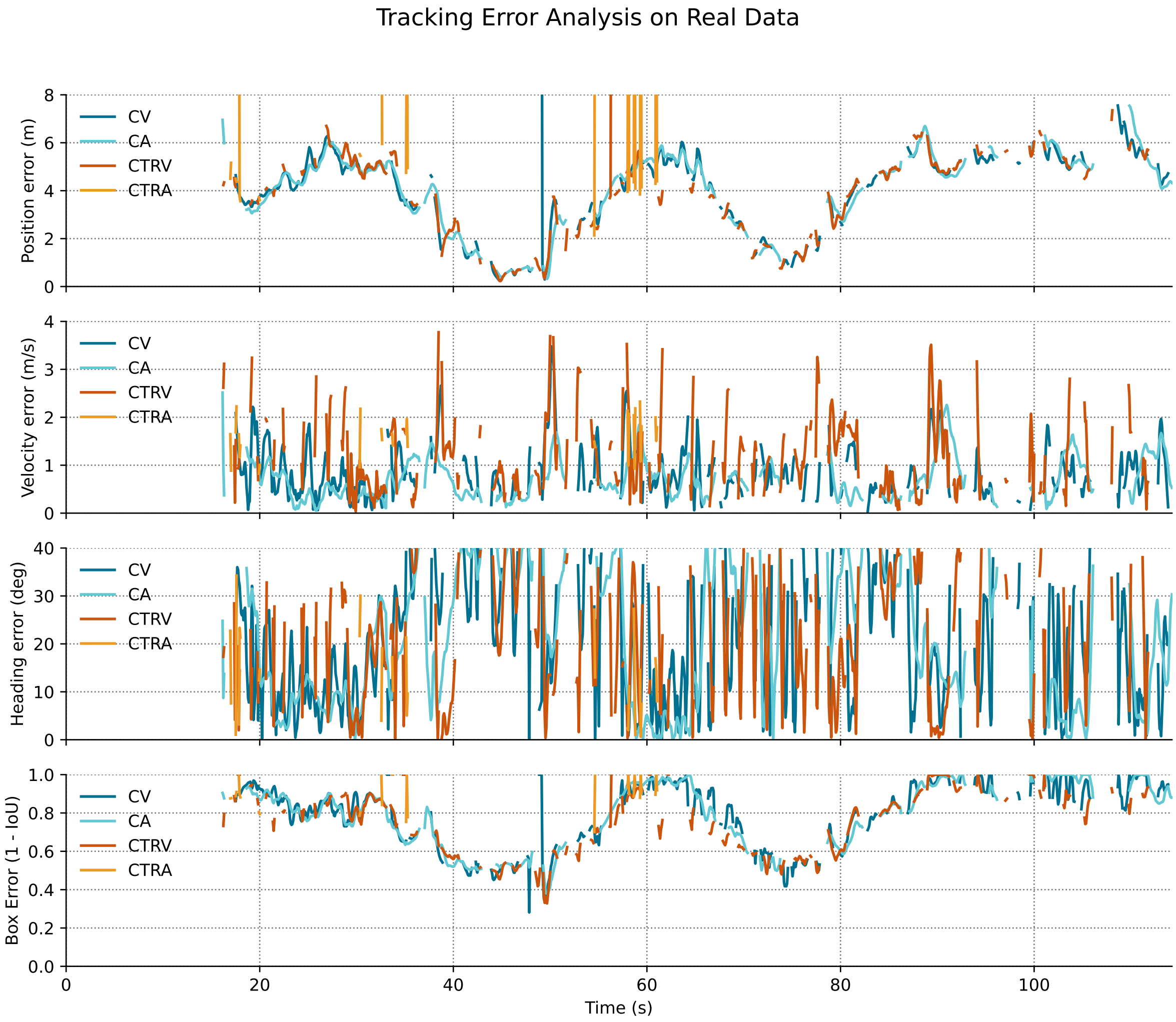


**Figure 4.9:** Tracking error comparison on the real-world trajectory. The subplots show positional, heading, velocity and bounding-box error over time for the indicated motion model.

The analysis of the real-world vessel trajectory, captured in Figure 4.7a, presents a significantly more challenging tracking problem than the simulated case. The corresponding error metrics are detailed in Figure 4.9. An immediate observation is the substantial increase in measurement noise across all metrics, with the heading error appearing particularly erratic. This is to be expected, as the vessel's true motion does not adhere to any single, simplified model; it is subject to complex

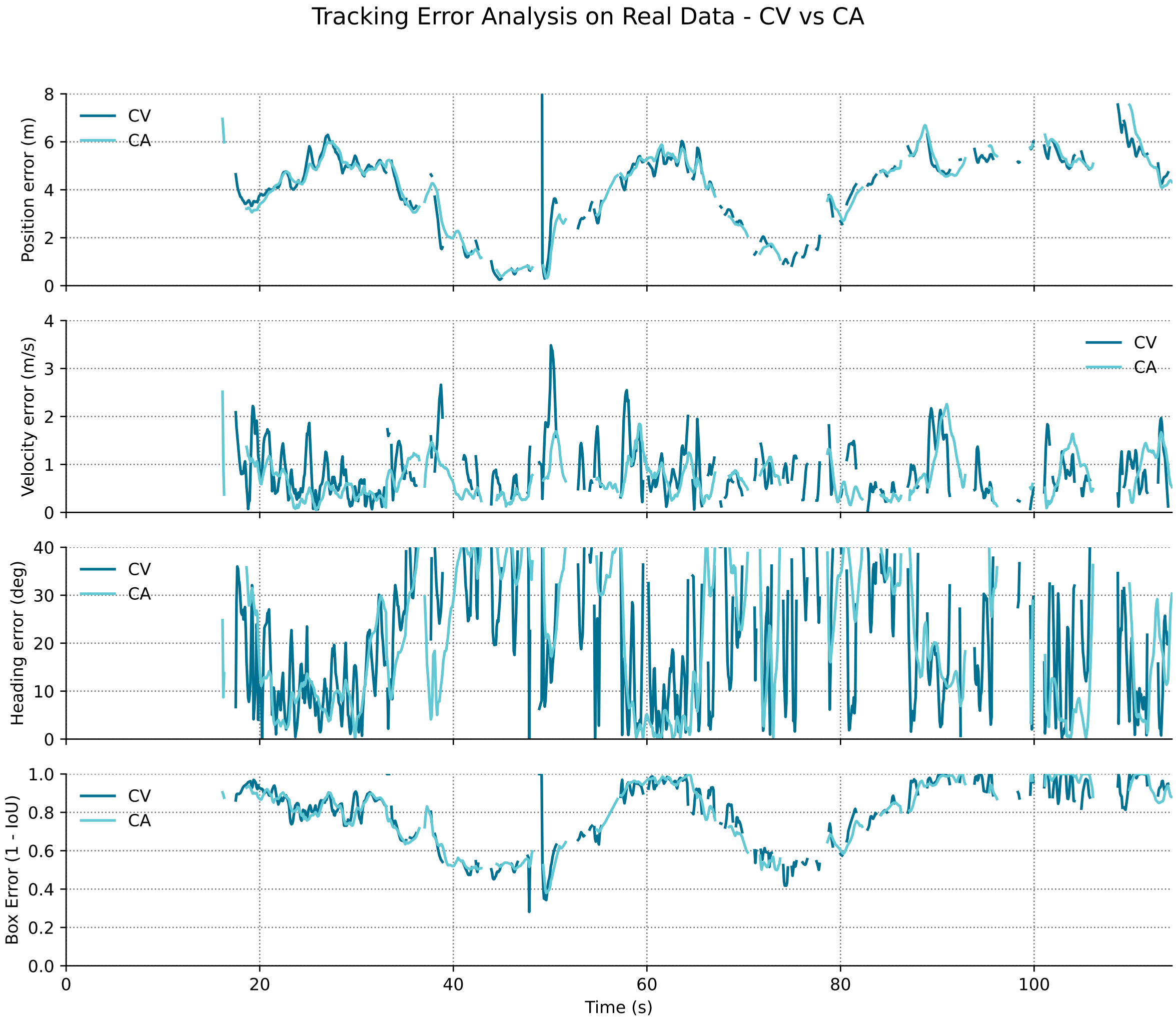


**Figure 4.10:** Tracking error comparison on the real-world trajectory focussing on just CV and CA. The subplots show positional, heading, velocity and bounding-box error over time for the indicated motion model.

dynamics, including side-slip, as it executes a very tight turn relative to its length.

A key consequence of this complexity is that track continuity is intermittent. The plots show frequent gaps, indicating that the target is either lost and later reacquired, or that the filter's state estimate diverges too far from the ground truth to be correctly associated. The degree of this intermittency varies between models. As highlighted in the focused comparison in Figure 4.10, the CA model provides a more consistent track than the CV model, maintaining the track for a longer duration, as indicated by the more extensive plot line.

Once again, the CA model outperforms the non-linear models, particularly in the estimation of velocity. This result is interesting, as one might expect the turn-based models to excel in this scenario. A potential explanation for their failure lies in the severity and speed of the manoeuvre. The vessel shifts from turning in one direction to the other very rapidly, which would require a sudden and large change in the filter's turn-rate state. The process noise associated with the turn rate may not be tuned to accommodate such rapid changes, preventing the filter's covariance from expanding quickly enough for it to adapt with sufficient agility. The CA model, lacking this specific turn-rate state, may be able to represent the manoeuvre more effectively through its linear acceleration components.

### 4.3.3. EKF vs UKF for Non-Linear Models

A direct comparison was made between the Extended Kalman Filter and the Unscented Kalman Filter for implementing the non-linear CTRV and CTRA motion models. The UKF was implemented according to the standard formulation. For all filter variants, the key parameters, namely the initial state covariances and the process noise terms, were optimised over at least 300 trials to ensure a fair comparison based on their best achievable performance.

The results for the simulated circular trajectory are presented in Figure 4.11. A distinct difference in initialisation performance is immediately apparent. The EKF exhibits significant initial instability, with large swings in error across all metrics, suggesting a prolonged convergence period. In contrast, the UKF achieves a stable and accurate estimate much more rapidly. However, it is important to note that once the EKF's initialisation phase is complete and its estimate has stabilised, its performance becomes remarkably similar to that of the UKF. Ultimately, both filter implementations for a given model converge to a highly accurate state estimate, for both position and shape. A persistent characteristic of both non-linear filters, however, is a slightly noisy and oscillatory heading estimate, a behaviour that is less pronounced in the linear CV and CA models.

When applied to the more demanding real-world data, shown in Figure 4.12, the performance dynamic between the two filters shifts. Interestingly, the EKF appears to handle certain aspects of this complex trajectory better than the UKF. Specifically, for the CTRA model, the EKF maintains a more consistent track during the initial section of the manoeuvre compared to its UKF counterpart. The precise reason for this reversal in performance is not immediately clear from the results, but it remains a noteworthy observation, highlighting that the theoretically more robust UKF is not universally superior in all practical scenarios.

## 4.4. Extent Model Performance Analysis

This section presents an analysis of the various extent models, focusing on their performance with real-world data. To illustrate their behaviour, zoomed-in sections from different time steps of the tracking algorithm are used, as depicted in Figure 4.13 and Figure 4.14. These figures showcase how the estimates evolve over time and are affected by incoming sensor data.

### 4.4.1. Random Matrix Model

The Random Matrix models, both the Tracking-by-Detection and Probability Hypothesis Density filter variants, demonstrate rapid convergence to a stable and accurate bounding box of the target. As seen in Figure 4.13a and Figure 4.13b, the elliptical estimates conform well to the target's shape. A notable advantage is the model's ability to skew the ellipse to cover the target effectively, even when the initial velocity estimate is offset. This characteristic is particularly beneficial for accommodating phenomena such as vessel side-slip.

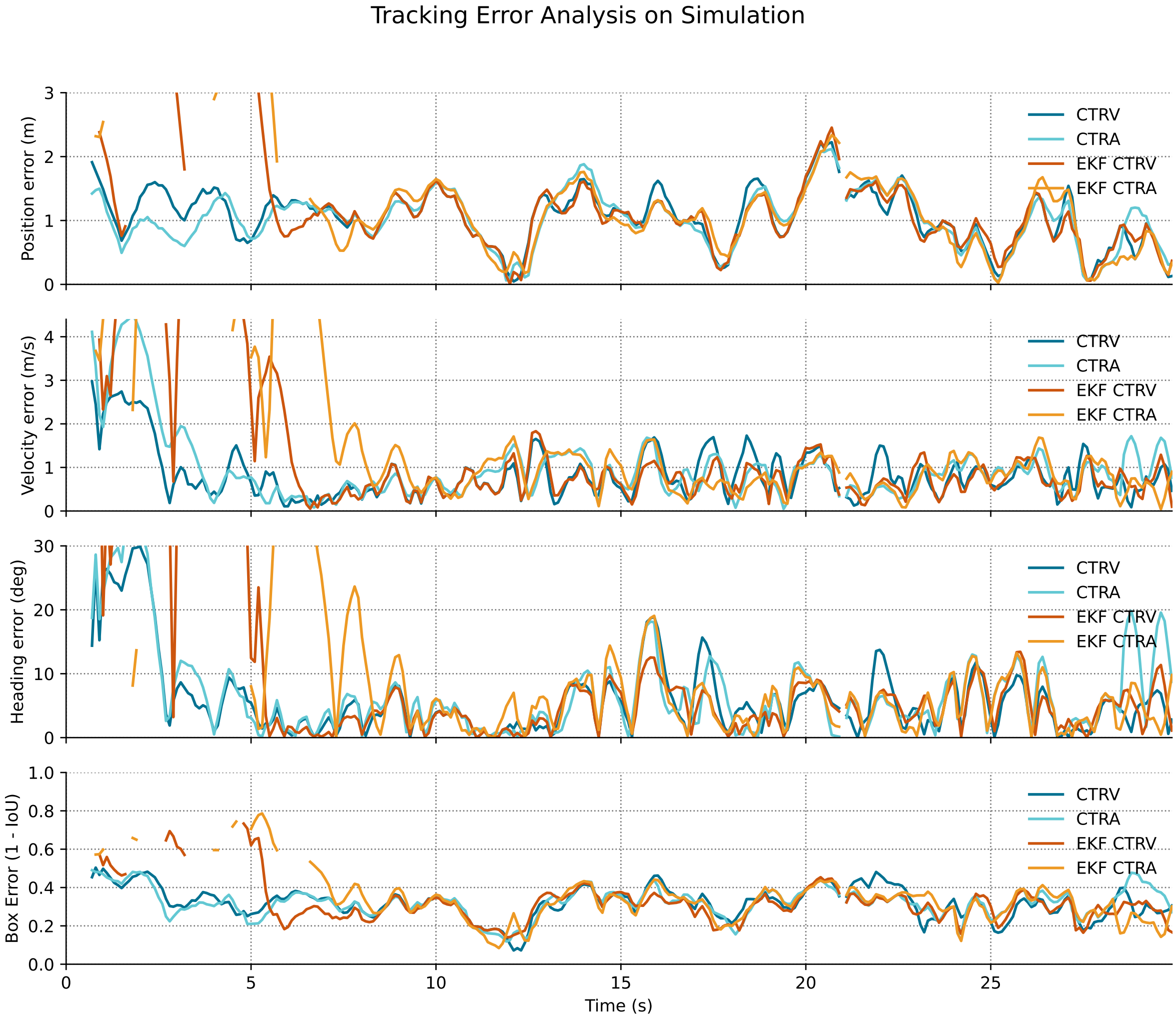


**Figure 4.11:** Tracking error comparison on the simulated circular trajectory for non-linear motion models implemented with EKF and UKF. Subplots show positional, heading, velocity and bounding-box error over time for each motion model and filter variant.

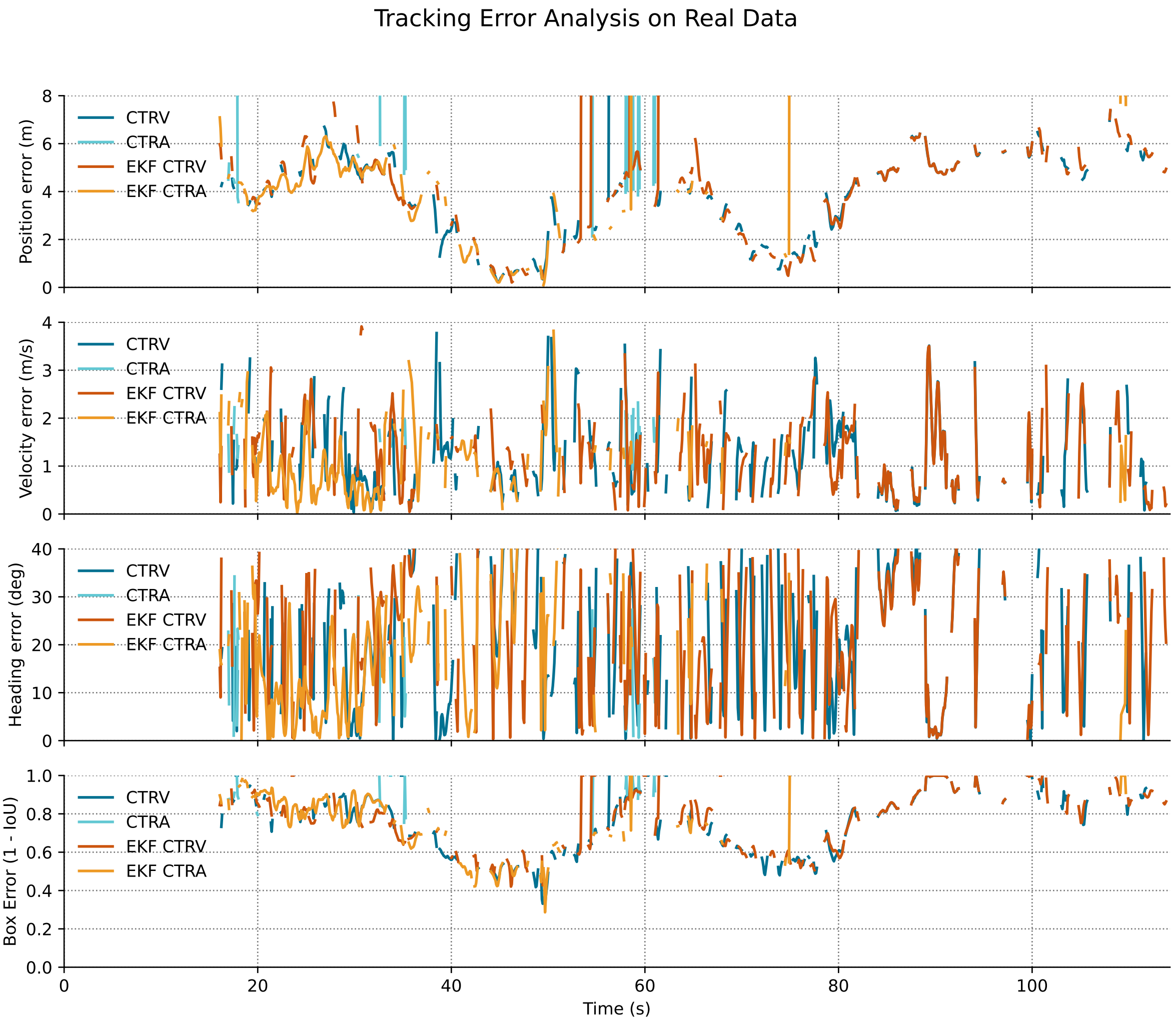
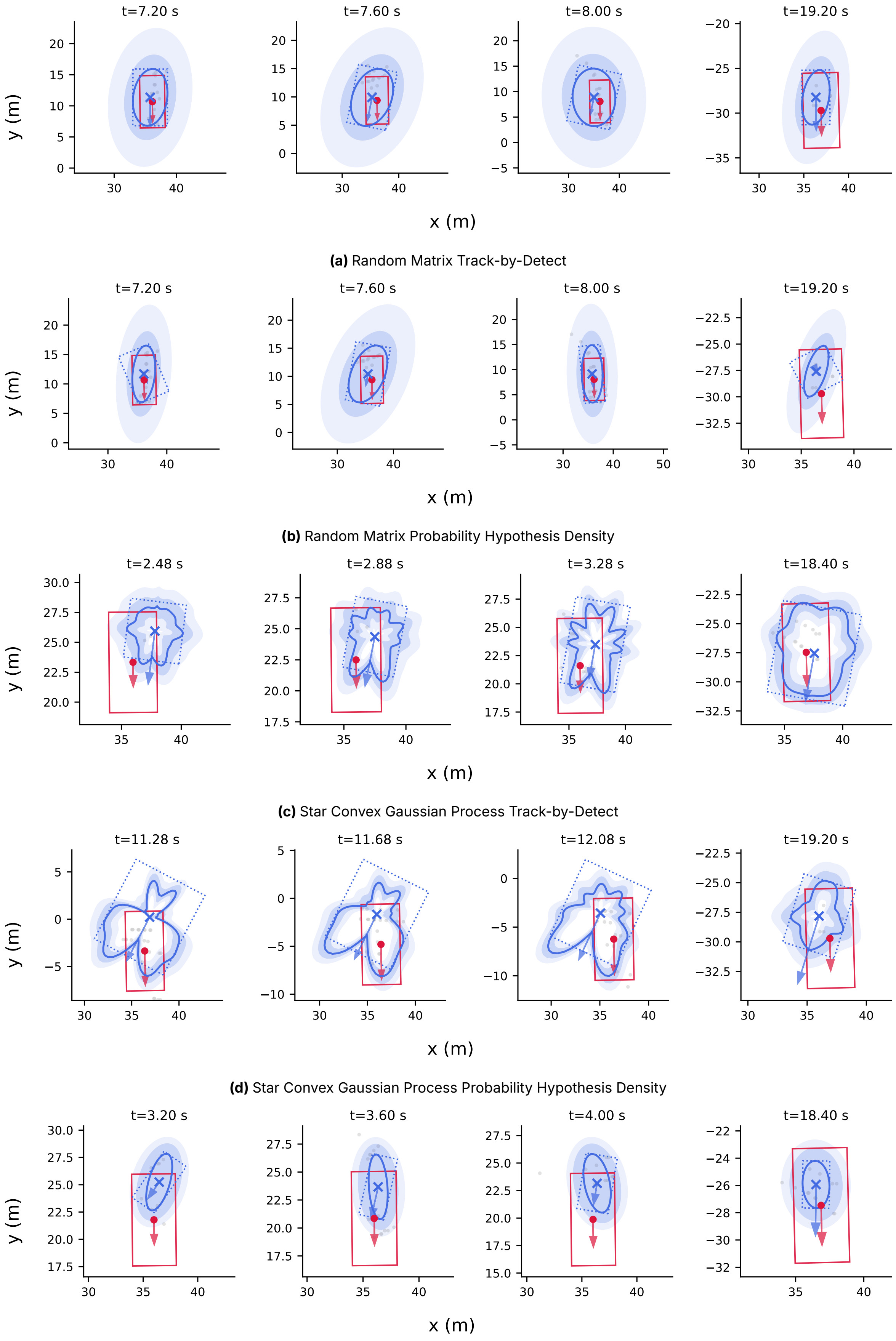


**Figure 4.12:** Tracking error comparison on the real-world vessel trajectory for non-linear motion models implemented with EKF and UKF. Subplots show positional, heading, velocity and bounding-box error over time for each motion model and filter variant.



**(a)** Random Matrix Track-by-Detect

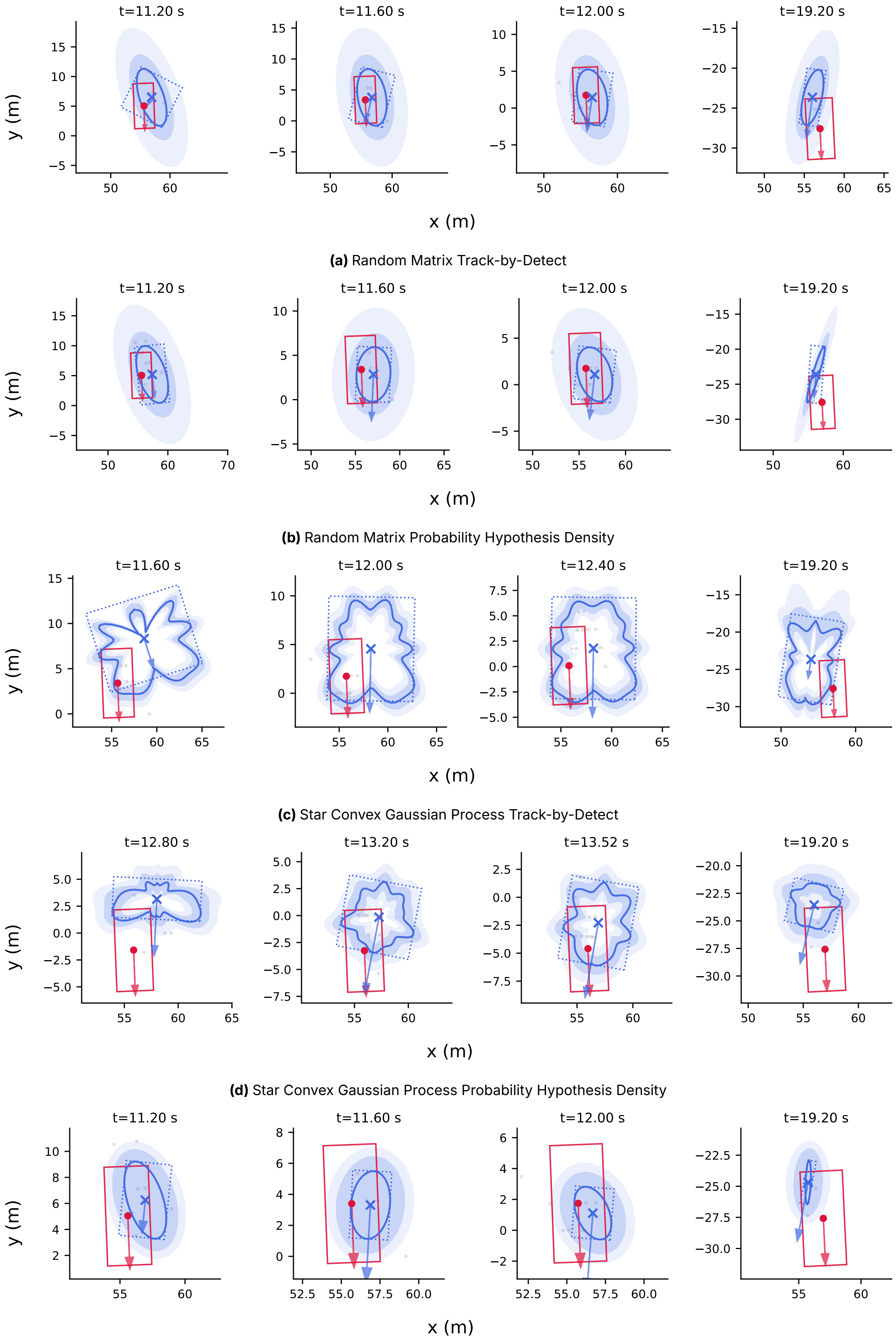
**(b)** Random Matrix Probability Hypothesis Density

**(c)** Star Convex Gaussian Process Track-by-Detect

**(d)** Star Convex Gaussian Process Probability Hypothesis Density

**(e)** Principle Axis Track-by-Detect

**Figure 4.13:** Comparison of extent models zoomed-in view at comparable time steps for target 0. Note that the time steps are not exactly the same between models due to differing quality of the tracks.



**(a)** Random Matrix Track-by-Detect

**(b)** Random Matrix Probability Hypothesis Density

**(c)** Star Convex Gaussian Process Track-by-Detect

**(d)** Star Convex Gaussian Process Probability Hypothesis Density

**(e)** Principle Axis Track-by-Detect

**Figure 4.14:** Comparison of extent models zoomed-in view at comparable time steps for target 1. Note that the time steps are not exactly the same between models due to differing quality of the tracks.

Across the dataset, minimal performance difference is observed between the TBD and PHD filter implementations. Both methods converge swiftly from an initial estimate to a stable one after only a few measurement updates. However, it is noted that there can be a slight delay before an initial track is formed.

When a target begins to leave the sensor's field of view, the model's estimate is observed to shrink and re-centre around the few remaining measurement points, as shown in the rightmost plots of Figure 4.14. This behaviour is expected, as the implemented filter lacks specific logic to account for objects exiting the scene. Nevertheless, compared to other models, the RM approach maintains its shape integrity more effectively during this phase.

### 4.4.2. Star-Convex Gaussian Process Model

The Star Convex Gaussian Process model exhibits distinct characteristics. Its extent function can expand asymmetrically, allowing it to extend further in one direction than another. This can lead to an initial estimate that trails behind the target while simultaneously elongating forwards to encompass all measurement points, as illustrated in Figure 4.13c. Over time, the estimate typically contracts and re-centres, but this initial elongation effect is a common feature.

A consistent observation is that the SCGP model often overestimates the target's size and struggles to accurately represent long, thin objects. In certain specific cases, it can provide a more stable or correct shape estimate than other models, but this comes at a significantly higher computational cost.

The combination of the SCGP extent model with the PHD filter (Figure 4.13d) appears to be suboptimal in this implementation. This pairing tends to produce oversized and irregularly shaped estimates rather than converging towards a well-defined rectangular shape. This may be attributable to modelling choices, such as the birth model, which might be less suitable for this particular filter configuration. Furthermore, there appears to be a tendency within the filter to expand the perimeter function rather than adjusting the kinematic state to better fit the measurements, although the estimate does generally improve over time.

### 4.4.3. Principal Axis Model

The Principal Axis model, which explicitly estimates the ellipse parameters rather than using a random matrix, frequently underestimates the size of the target (see Figure 4.13e). It also demonstrates a high sensitivity to new data, readily adjusting its estimate rather than maintaining a more stable, time-averaged shape. While this reactivity can be tuned, the parameters used here were selected following a parameter study that identified them as optimal for this dataset.

In this work, the PA model failed to produce any valid estimates when combined with the PHD filter for the targets present in the data.

Similar to the other models, the PA estimate shrinks considerably as the target leaves the field of view, adjusting to fit the few detections originating from the corner of the object (visible in Figure 4.14e). This behaviour again highlights the model's high responsiveness to the immediate measurement set.

# 5 Discussion

This chapter interprets the results from Chapter 4, focusing on the trade-offs between algorithmic complexity and tracking performance in the maritime domain. It synthesises the key findings to discuss the comparative performance of the different filters, shape models, and motion models tested. The discussion also covers practical challenges, such as hyperparameter optimisation and sensor limitations.

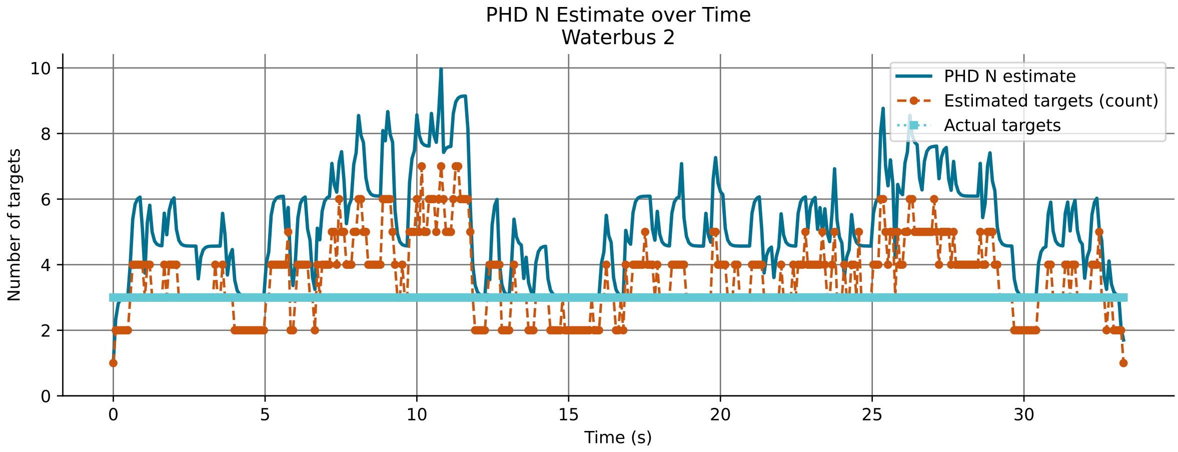
## 5.1. Filter Frameworks: A Comparison of TBD, PHD, and PMBM Approaches

A key finding from this work is that increased filter complexity did not necessarily yield superior tracking performance. In fact, the most effective and reliable results were achieved with the comparatively simpler frameworks: a baseline Tracking-by-Detection implementation and the Probability Hypothesis Density filter. Both of these methods performed exceptionally well when paired with the Random Matrix (GIW) extent model. On simulated data, this effectiveness was particularly evident, with the state estimates of both filters converging to low error values and maintaining a stable number of target tracks with very few false positives, aligning with the expectations set during the parameter study.

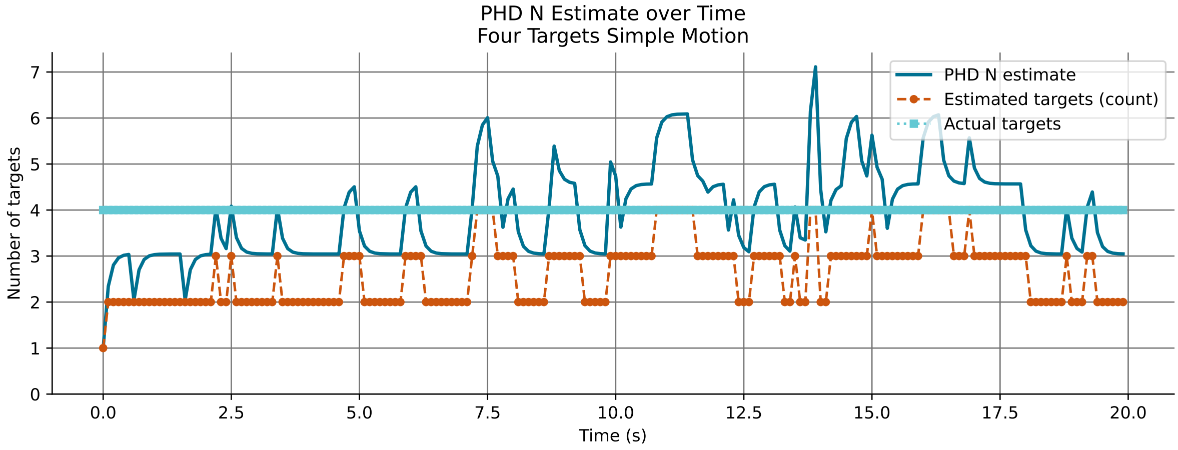
In contrast, the more sophisticated Poisson Multi-Bernoulli Mixture filter, even when combined with the same Random Matrix extent model, struggled to produce comparable results. While it was capable of forming tracks on the targets, its output was plagued by a significantly higher number of false positives. Furthermore, the tracks it generated were often unstable, appearing and disappearing intermittently. This unreliable performance is particularly noteworthy given that the PMBM filter is far more computationally intensive, requiring techniques such as gating merely to be a tractable solution. The results suggest a clear mismatch between its theoretical sophistication and its practical utility in this specific application.

The PHD filter occupies a middle ground in terms of complexity and performance. It performed on par with the simple TBD baseline, providing stable and accurate tracks. The primary theoretical advantage of the PHD filter is its inherent ability to estimate the number of targets (the cardinality) directly from the integral of the density function over the entire state space. This is a more principled approach than simply counting the number of confirmed tracks after state extraction. However, despite this conceptual elegance, the PHD filter did not deliver a markedly better performance than the baseline single-target tracker with a simple assignment logic, as can be seen in Figure 5.1. The additional implementation and computational overhead did not translate into a tangible improvement in the final tracking output for this scenario.

To put this in additional context Figure 5.2 shows the average computation time per frame for each filter configuration. The PMBM filter is orders of magnitude slower than both the PHD and TBD approaches, which are comparable in speed. While this does come with the side



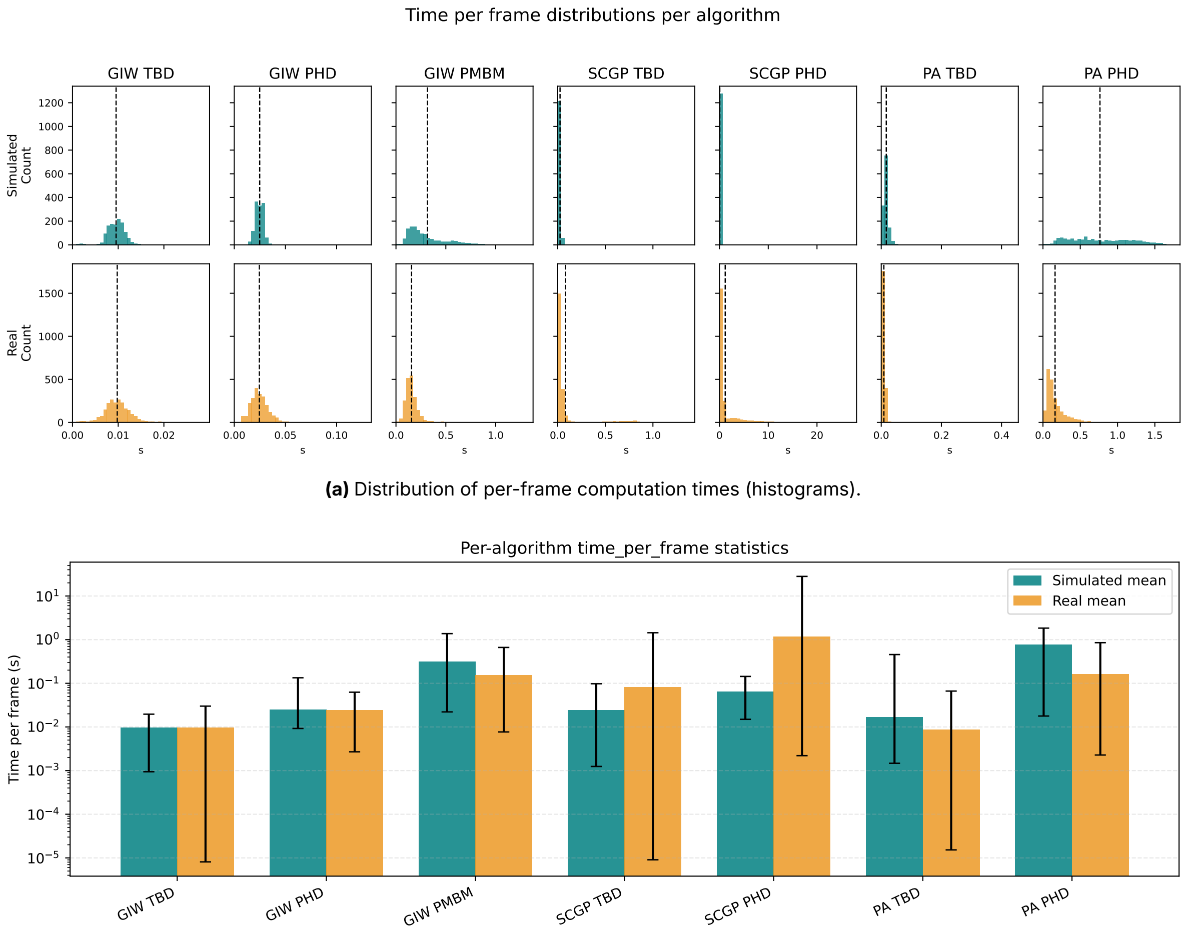
**(a)** Single-target (waterbus_2) scenario



**(b)** Four-target simulated scenario

**Figure 5.1:** Estimated number of targets produced by the PHD filter over time for two qualitative cases: (a) the waterbus_2 recording, and (b) a four-target scene. These plots illustrate cardinality estimation behaviour, including stability and common failure modes (misses and false positives).

note of none of these algorithms being optimized for speed and efficiency in this work beyond what was required to make them computationaly viable (i.e. gating, pruning, etc.). This stark difference in computational demand further emphasises the impracticality of the PMBM filter for real-time applications in this context, especially when its performance does not justify the additional complexity. What is additionally noteworthy is that the choice of extent model and it's implementatation can greatly effect the computation time, with the non-linear models (SCGP and PA) being significantly slower compared to the linear Random Matrix(GIW) model across all filter types. Even worse then one would expect when considering the modest increase of the Random Matrix model from TBD to PHD.



**(a)** Distribution of per-frame computation times (histograms).

**(b)** Average computation time per frame (bar chart) for each filter configuration. Note the logarithmic scale on the y-axis.

**Figure 5.2:** Stacked visualisation of computation-time results: (a) per-frame time distributions and (b) mean time per frame per filter configuration. Note that these figures only serve as an indication of relative computational cost, as the implementations were not optimised for speed beyond basic techniques to ensure tractability.

## 5.2. The Diminishing Returns of Complex Motion Models

Complex motion models offered surprisingly little benefit in this application. The experimental results consistently showed that the non-linear Constant Turn Rate and Velocity and Constant Turn Rate and Acceleration models offered no significant performance advantage over the much simpler linear Constant Velocity and Constant Acceleration models. This observation can be explained by three main factors.

Low Vehicle Dynamics: Vessels in inland waterways, due to their large mass and operational constraints, exhibit low dynamics. They accelerate, decelerate, and turn slowly and predictably, so the abrupt, highly non-linear manoeuvres that would necessitate a model like CTRA are simply not prevalent.

High Sensor Update Rate: The 4D radar sensor provides measurements at a high frequency. The short time interval between updates means that a simple linear extrapolation of the state is often a sufficiently accurate prediction, as the vessel's displacement and change in heading between frames are small, reducing the need for a more sophisticated curvilinear model.

Dominance of Sensor Noise: At longer ranges, the inherent measurement noise and sparsity of the radar point cloud are significant, as can be clearly seen in Figure H.10d. Any subtle accuracy gains afforded by a more precise motion model are likely swamped by the uncertainty in the measurement data itself, so the filter's primary challenge becomes noise rejection rather than precise motion prediction. This finding is context-dependent; the models' utility was limited by the low dynamics of the vessels and the high sensor update rate. Their additional complexity and computational overhead are therefore not justified in this context, though they may prove more valuable for faster targets or different sensor configurations.

### 5.2.1. Interaction with Extent Modelling

One might argue that a key advantage of a non-linear model is its ability to decouple an object's heading from its velocity vector. This is particularly relevant for vessels performing slow manoeuvres, such as docking, where they can exhibit significant side-slip (i.e., moving sideways whilst pointing forwards). While a non-linear motion model could explicitly estimate this, the use of an extent model in this framework provides an alternative and effective solution.

The extent model, by its nature, estimates the shape and orientation of the vessel separately from its kinematic state. It can therefore represent a side-slipping vessel accurately without requiring a complex motion model to capture the non-linear dynamics. In effect, the task of representing the vessel's orientation is offloaded to the extent model. While this approach means that the dynamic interplay between heading and velocity (e.g., skidding during a turn) is not explicitly estimated, the primary goal of determining the space occupied by the vessel—its 'risk shape'—is still achieved.

Ultimately, the combination of a simple linear motion model, such as Constant Acceleration, with a flexible extent model, like the Random Matrix model, provides the best of both worlds for this application. Given the high sensor update rate, this pairing yields a stable and accurate track that can estimate the hazardous area around other vessels, fulfilling the main objective of the tracking system without incurring the unnecessary complexity and computational overhead of non-linear motion models.

## 5.3. On Extent Modelling Approaches

The performance of the various extent models demonstrates a clear advantage for linear models within the implemented tracking frameworks. Of the three models investigated, the Random Matrix, the Star-Convex Gaussian Process, and the Principal Axis model, only the Random Matrix approach constitutes a fully linear system. Both the SCGP and Principal Axis models are inherently non-linear, necessitating the use of an Extended Kalman Filter for state estimation; a process that requires the derivation and computation of Jacobian matrices for linearisation.

### 5.3.1. The Impracticality of Complex Shape Models

Although sophisticated models such as the SCGP exhibit theoretical promise, their practical benefits were not realised within the scope of this application. Given a sufficient density of high-quality measurements, it is conceivable that a Gaussian Process could estimate a more precise and complex object shape. However, for the primary objective of determining a robust and simple extent for applications such as collision avoidance, the additional complexity offers no tangible advantage. A simpler model that estimates an elliptical extent, such as the Principal Axis or Random Matrix model, provides sufficient information for this purpose. Consequently, the computational overhead and tuning effort required for the non-linear SCGP model are not justified by its performance in this context.

### 5.3.2. The Impact of Non-Linearity in Elliptical Estimation

A particularly insightful finding is the performance discrepancy between the Random Matrix and Principal Axis models. Although both models estimate an ellipse that encompasses the target's measurements, the Random Matrix model performed significantly better. This disparity is attributable to the non-linearity inherent in the Principal Axis formulation. The explicit estimation of orientation is a non-linear problem, which introduces complexities and potential instabilities into the filter that are absent in the purely linear Random Matrix system.

This suggests that the tracking frameworks, from the baseline tracker to the more advanced Random Finite Sets filters, function most effectively when their constituent components are linear. These systems are built upon a foundation of probability theory that is most robust and tractable for linear-Gaussian systems. Although the Principal Axis model appears simpler superficially, estimating only a few intuitive parameters, it proves more challenging to handle in practice, as the non-linear mapping from measurements to the state is less direct and stable.

In contrast, the Random Matrix model benefits from its strong probabilistic grounding. It provides a natural and mathematically sound method for associating measurements with a state estimate via the Mahalanobis distance. Furthermore, it possesses well-defined and analytically tractable methods for operations such as merging, which are crucial within multi-target filtering. These properties, namely its linearity and rigorous probabilistic formulation, are the likely reason for its prevalence and efficacy as a model in the mathematical literature on extended object tracking.

## 5.4. Practical Challenges and System-Level Limitations

Beyond the core algorithmic comparison, the study highlighted several practical challenges inherent to the application. These challenges span from computational constraints and data quality issues to simplifying assumptions made about the operating environment. The following sections detail these limitations, providing context for the interpretation of the results and outlining avenues for future research.

### 5.4.1. Hyperparameter Optimisation on Real-World Data

A significant challenge was the failure of the automated hyperparameter optimisation process on the real-world dataset. As noted in the results, the optimiser tended towards solutions that produced zero detections. This counter-intuitive result likely stems from the interaction between the GOSPA metric and the nature of real-world radar data, which contains significantly more clutter and potential for false positives than the simulation. The GOSPA metric penalises both missed targets and false tracks. In a cluttered environment, a poorly tuned filter can generate a large number of false tracks, leading to a very high GOSPA score. The optimiser may then discover that the easiest way to reduce this high penalty is to become extremely conservative, raising detection thresholds to the point where it reports no targets at all. In this state, the penalty is simply the cost of the missed ground truth targets, which can be lower than the cost of tracking the ground truth targets plus many false positives. This suggests that the cost function landscape is non-trivial and that for real-world data, a more sophisticated optimisation strategy or a different evaluation metric might be required to avoid such local minima.

### 5.4.2. Constrained Scope of Algorithmic Permutations

A primary limitation of this research was the inability to test every possible permutation of the tracking filters, motion models, and extent models. The combinatorial explosion of potential configurations, coupled with significant time and computational constraints, necessitated a focused approach. The core of the comparative analysis was therefore structured to evaluate the different filter and extent model combinations against a common baseline: the linear Constant Velocity (CV) motion model.

This decision meant that the potential benefits of pairing certain filters with more complex, non-linear motion models were not explored. For instance, the Gaussian Process (GP) motion model, which can infer and utilise target heading information from the point cloud, might have yielded superior performance in combination with the PHD or PMBM filters, particularly during target manoeuvres. However, these combinations were not tested. Furthermore, the PMBM filter, owing to its substantially higher computational demand, was only evaluated with the CV motion model and the Random Matrix extent model. An exhaustive optimisation and comparison of the PMBM filter with other non-linear or extent models was computationally intractable within the project's timeframe. Consequently, while the study offers a clear comparison of the algorithms under linear assumptions, it cannot make definitive conclusions about their relative performance in highly non-linear scenarios where alternative motion models might have a distinct advantage.

### 5.4.3. Ground Truth Generation and Data Fidelity

The generation of accurate ground truth for real-world, uncontrolled maritime environments presented a persistent challenge that directly impacts the certainty of the performance evaluation. The fidelity of the ground truth varied significantly across the datasets due to several factors:

#### Unaccounted-for Objects and Clutter

The real-world scenes contained numerous radar returns that did not correspond to any ground truth target. These included reflections from static objects like channel markers and buoys, as well as detections from other small vessels not included in the ground truth. This unlabelled clutter complicates evaluation, as a filter might generate a valid track for a real object that is simply absent from the ground truth data, unfairly penalising the algorithm's performance score.

#### Inconsistent Data Sources

The ground truth was compiled from multiple sources of varying quality. For some vessels, ground truth was derived from sparse AIS data, which often provided position updates only once every several seconds. This necessitated interpolation between a limited number of data points to form a continuous trajectory. For two small pleasure craft, trajectories and extents had to be manually annotated by inspecting the point cloud data frame-by-frame—a subjective and labour-intensive process.

#### High-Fidelity but Context-Limited Data

The highest quality ground truth was a GPS track used for the motion model analysis. While this provided a highly accurate trajectory for a single target, the surrounding scene was densely populated with other untracked objects. This made the dataset unsuitable for a fair multi-target tracking evaluation, restricting its use to an isolated analysis of motion model performance.

These limitations are representative of the difficulties encountered in real-world system deployment and introduce a degree of uncertainty into the quantitative results.

### 5.4.4. System-Level Simplifications and Implementation Nuances

To maintain a tractable research scope, several simplifying assumptions were made regarding the system and its operational environment. While necessary, these simplifications limit the direct applicability of the findings to a real-world, dynamic deployment.

#### Stationary-sensor assumption

A core simplification was the stationary sensor assumption. All data was collected and processed from a fixed-position radar, which eliminates the need for ego-motion compensation. In a practical application, such as on a moving vessel, the sensor's own motion would add significant complexity to the state estimation problem. The performance of the motion models, in particular, may have been artificially improved by this static context.

#### Edge effects and track initiation/termination

Furthermore, the study neglected edge effects, where targets are only partially within the sensor's field of view. No specific logic was implemented to handle the initiation or termination of tracks for objects entering or leaving the scene. A robust, deployed system would require dedicated logic for this, or potentially a network of multiple sensors to provide a complete and gapless surveillance area.

### Measurement model simplification

The measurement model was also simplified. The focus was on comparing different extent models, but the underlying sensor noise was modelled as simple Cartesian noise. A more sophisticated non-linear measurement model that accounts for sensor characteristics—such as the increase in angular uncertainty with distance—was not included. While this could be addressed in future work by incorporating a non-linear noise covariance matrix, it remains a limitation of the current study.

### Implementation anomaly: PHD filter with GP motion model

Finally, an implementation anomaly was observed in the combination of the PHD filter and the GP motion model. While both components were verified to function correctly in isolation, their combined performance was significantly lower than anticipated. This suggests an unexpected negative synergy, possibly arising from the calculation of the log-likelihoods within the filter update step. Although the implementation was functional, its sub-optimal performance highlights the subtle but significant challenges of integrating complex probabilistic models.

# 6 Conclusions and Recommendations

## 6.1. Summary of Findings

This thesis set out to investigate the problem of multiple Extended Target Tracking for maritime vessels using 4D millimetre-wave radar point clouds. The central aim was to examine the balance between algorithmic sophistication and model fidelity in the context of multi-object tracking, with a specific focus on the comparative performance of three core tracking paradigms: a baseline Tracking-by-Detection (TBD) approach, the Probability Hypothesis Density (PHD) filter, and the Poisson Multi-Bernoulli Mixture (PMBM) filter. In doing so, the work addressed the research questions posed in Chapter 1, namely how different combinations of filter frameworks, motion models, and extent models influence tracking performance, and what level of model complexity is warranted for robust maritime state estimation.

The results presented in Chapter 4 and discussed in Chapter 5 reveal several key insights. Simpler paradigms generally outperformed their more sophisticated counterparts in terms of stability and robustness. Both the TBD and PHD approaches delivered consistent performance, whereas the PMBM filter, despite its theoretical optimality within Random Finite Set theory, proved computationally burdensome and prone to unstable track management. Complex non-linear motion models (CTRV, CTRA) yielded negligible improvements over simpler linear models (CV, CA) given the low dynamics of maritime vessels and the high update rate of the radar sensor. Regarding extent models, the Random Matrix (GIW) model consistently achieved the best balance between accuracy, stability, and computational efficiency; more elaborate models (SCGP, PA) often struggled with stability, over-fitting, or excessive computational requirements.

## 6.2. Revisiting the Research Questions

While the complete answers to the research questions are elaborated in Chapter 5, the key conclusions can be summarised as follows:

### 6.2.1. How do different motion models and state estimation filters influence tracking performance?

Simple linear motion models (CV, CA) are generally sufficient for accurate tracking of maritime vessels; complex non-linear models (CTRV, CTRA) provide little measurable benefit. While PMBM is theoretically rigorous, in practice TBD and PHD achieved comparable or superior results at a fraction of the computational cost. Thus, maritime tracking performance depends less on algorithmic complexity and more on robust, well-tuned simpler models.

### 6.2.2. What is the optimal balance between algorithmic complexity and motion model fidelity?

The optimal balance is achieved by pairing simple linear motion models with the Random Matrix extent model. This configuration provided accurate cardinality estimation, stable state tracking, and reliable extent estimation, outperforming more complex alternatives in both simulated and real-world trials. Parsimonious models that prioritise interpretability, robustness, and computational efficiency are recommended.

## 6.3. Practical Challenges and Limitations

The work identified several system-level limitations. Hyperparameter optimisation, while effective in simulation, was less reliable in real-world experiments and sometimes converged to trivial or overly conservative configurations. The experiments used static sensors, excluding complications introduced by sensor ego-motion. Although approximation techniques mitigated some computational issues of the PMBM filter, the framework remains impractical for real-time deployment on typical maritime hardware.

## 6.4. Directions for Future Research

The research presented in this thesis establishes a foundation for 4D radar-based extended object tracking in maritime environments. However, the study's deliberate constraints highlight several promising avenues for future investigation that would enhance the system's capabilities, robustness, and practical applicability. These directions span platform dynamics, sensor fusion, algorithmic sophistication, and implementation challenges.

### 6.4.1. Dynamic Platforms and Field of View Limitations

A primary extension is the transition from a stationary to a dynamic sensor platform. The current work was confined to a static sensor, which simplifies motion modelling. Future research should address the significant challenges introduced by platform ego-motion. This will necessitate the integration of robust ego-motion compensation techniques to distinguish between the sensor's movement and the true motion of observed targets. Investigating the performance of the non-linear models presented herein under these conditions will be critical, as the additional non-linearities introduced by platform motion may exceed their effective operational envelope.

Closely related is the need to address the limitations imposed by the sensor's narrow field of view. This study revealed that estimation accuracy degrades when vessels are only partially observed at the periphery of the sensor's view. A key area for future work is the development of strategies to handle these partial detections. This could be achieved by creating a composite, wider field of view through the fusion of data from multiple, strategically placed radar sensors. Alternatively, fusing the 4D radar with complementary sensors, such as LiDAR or cameras, could provide the additional information required to maintain accurate state and extent estimates for partially observed targets.

### 6.4.2. Advanced Detection and State Estimation

Further advancements could be realised by moving beyond the baseline detection algorithm employed in this work. The simple DBSCAN algorithm served its purpose but is agnostic to the sensor's polar coordinate system and the specific morphology of maritime targets. Future work should explore more sophisticated detection methods. For instance, a grid-based clustering algorithm operating on a polar grid could yield more consistent detections. An even more transformative approach would be to employ 3D object detectors capable of generating bounding box estimates directly from the radar point cloud. This would fundamentally reframe the problem from extended object tracking to the tracking of bounding boxes, potentially simplifying the state estimation process and improving robustness, especially given the observed performance gap between theoretical and real-world data in this study.

Furthermore, this research did not fully exploit the richness of the available Doppler measurements. A significant opportunity exists to better integrate this velocity information into the tracking framework. Future research could investigate using Doppler data within the filter's update step to directly refine the target's velocity estimate. More advanced techniques could even leverage Doppler measurements from multiple points across an extended target to estimate an intra-object velocity gradient, providing insights into its rotational motion (yaw rate), which is invaluable for predicting manoeuvres.

### 6.4.3. System Robustness and Practical Implementation

Finally, for any tracking system to be operationally viable, it must be robust and efficient. The current work highlighted a discrepancy between performance on idealised data and complex, real-world scenarios. Future research should therefore focus on developing adaptive hyperparameter optimisation strategies. Machine learning techniques could be leveraged to allow the tracking system to dynamically tune its parameters—such as process and measurement noise—in response to changing environmental conditions or target behaviours, thereby improving its real-world robustness.

Lastly, the real-time feasibility of these algorithms must be assessed. Future work should involve optimising the implementations for deployment on embedded maritime systems, where computational resources are often constrained. This involves a critical analysis of the trade-offs between estimation accuracy and computational load, ensuring the chosen algorithms can operate effectively within the demanding cycle times of a real-world navigation and surveillance system.

## 6.5. Final Reflection

In conclusion, robust maritime tracking with 4D radar does not require maximal algorithmic complexity. The most effective solutions arise from principled yet efficient models that balance accuracy, efficiency, and interpretability. This insight supports the development of reliable perception systems for autonomous and semi-autonomous maritime navigation, where practical deployment considerations are as critical as theoretical elegance.

# References

[1] S. Thombre et al., "Sensors and AI Techniques for Situational Awareness in Autonomous Ships: A Review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 64–83, Jan. 2022, ISSN: 1524-9050, 1558-0016. DOI: 10.1109/TITS.2020.3023957.

[2] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," *Artificial Intelligence*, vol. 293, p. 103 448, Apr. 2021, ISSN: 0004-3702. DOI: 10.1016/j.artint.2020.103448.

[3] K. Granstrom, M. Baum, and S. Reuter, *Extended Object Tracking: Introduction, Overview and Applications*, Feb. 2017. DOI: 10.48550/arXiv.1604.00970. arXiv: 1604.00970 [cs].

[4] K. Granström and M. Baum, "A Tutorial on Multiple Extended Object Tracking," *TechRxiv*, Feb. 2022. DOI: 10.36227/techrxiv.19115858.v1.

[5] X. Weng, J. Wang, D. Held, and K. Kitani, *3D Multi-Object Tracking: A Baseline and New Evaluation Metrics*, Jul. 2020. DOI: 10.48550/arXiv.1907.03961. arXiv: 1907.03961 [cs].

[6] Á. F. García-Fernández, J. L. Williams, K. Granström, and L. Svensson, "Poisson multi-Bernoulli mixture filter: Direct derivation and implementation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 4, pp. 1883–1901, Aug. 2018, ISSN: 0018-9251, 1557-9603, 2371-9877. DOI: 10.1109/TAES.2018.2805153. arXiv: 1703.04264 [cs].

[7] S. Yao et al., *WaterScenes: A Multi-Task 4D Radar-Camera Fusion Dataset and Benchmarks for Autonomous Driving on Water Surfaces*, Jun. 2024. DOI: 10.48550/arXiv.2307.06505. arXiv: 2307.06505 [cs].

[8] J. Hasch, E. Topak, R. Schnabel, T. Zwick, R. Weigel, and C. Waldschmidt, "Millimeter-Wave Technology for Automotive Radar Sensors in the 77 GHz Frequency Band," *IEEE Transactions on Microwave Theory and Techniques*, vol. 60, no. 3, pp. 845–860, Mar. 2012, ISSN: 1557-9670. DOI: 10.1109/TMTT.2011.2178427.

[9] E. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, Lake Louise, Alta., Canada: IEEE, 2000, pp. 153–158, ISBN: 978-0-7803-5800-3. DOI: 10.1109/ASSPCC.2000.882463.

[10] M. Kolat, O. Törő, and T. Bécsi, "Performance Evaluation of a Maneuver Classification Algorithm Using Different Motion Models in a Multi-Model Framework," *Sensors*, vol. 22, no. 1, p. 347, Jan. 2022, ISSN: 1424-8220. DOI: 10.3390/s22010347.

[11] D. Svensson, "Derivation of the discrete-time constant turn rate and acceleration motion model," in *2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Oct. 2019, pp. 1–5. DOI: 10.1109/SDF.2019.8916654.

[12] J. W. Koch, "Bayesian approach to extended object and cluster tracking using random matrices," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 3, pp. 1042–1059, Jul. 2008, ISSN: 1557-9603. DOI: 10.1109/TAES.2008.4655362.

[13] 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, Dec. 2015, ISSN: 1558-0644. DOI: 10.1109/TGRS.2015.2444794.

[14] Z. Liang, F. Liu, and J. Gao, "Improved GGIW-PHD filter for maneuvering non-ellipsoidal extended targets or group targets tracking based on sub-random matrices," *PLoS ONE*, vol. 13, no. 2, e0192473, Feb. 2018, ISSN: 1932-6203. DOI: 10.1371/journal.pone.0192473.

[15] N. Wahlstrom and E. Ozkan, "Extended Target Tracking Using Gaussian Processes," *IEEE Transactions on Signal Processing*, vol. 63, no. 16, pp. 4165–4178, Aug. 2015, ISSN: 1053-587X, 1941-0476. DOI: 10.1109/TSP.2015.2424194.

[16] J. S. Fowdur, M. Baum, and F. Heymann, "An Elliptical Principal Axes-based Model for Extended Target Tracking with Marine Radar Data," in *2021 IEEE 24th International Conference on Information Fusion (FUSION)*, Nov. 2021, pp. 1–8. DOI: 10.23919/FUSION49465.2021.9627039.

[17] J. S. Fowdur, M. Baum, F. Heymann, and P. Banys, "An Overview of the PAKF-JPDA Approach for Elliptical Multiple Extended Target Tracking Using High-Resolution Marine Radar Data," *Remote Sensing*, vol. 15, no. 10, p. 2503, Jan. 2023, ISSN: 2072-4292. DOI: 10.3390/rs15102503.

[18] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple Online and Realtime Tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sep. 2016, pp. 3464–3468. DOI: 10.1109/ICIP.2016.7533003. arXiv: 1602.00763 [cs].

[19] V. Stanojević and B. Todorović, *BoostTrack++: Using tracklet information to detect more objects in multiple object tracking*, Aug. 2024. DOI: 10.48550/arXiv.2408.13003. arXiv: 2408.13003 [cs].

[20] L. Rakai, H. Song, S. Sun, W. Zhang, and Y. Yang, "Data association in multiple object tracking: A survey of recent techniques," *Expert Systems with Applications*, vol. 192, p. 116 300, Apr. 2022, ISSN: 09574174. DOI: 10.1016/j.eswa.2021.116300.

[21] W. Wu, H. Sun, M. Zheng, and W. Huang, *Target Tracking with Random Finite Sets*. Singapore: Springer Nature Singapore, 2023, ISBN: 978-981-19-9814-0. DOI: 10.1007/978-981-19-9815-7.

[22] "A Theoretical Foundation for the Stein-Winter "Probability Hypothesis Density (PHD)" Multitarget tracking Approach," p. 21, Jun. 2000.

[23] R. Mahler, "PHD filters for nonstandard targets, I: Extended targets," in *2009 12th International Conference on Information Fusion*, Jul. 2009, pp. 915–921.

[24] 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, Oct. 2012, ISSN: 0018-9251. DOI: 10.1109/TAES.2012.6324703.

[25] K. Granström, M. Fatemi, and L. Svensson, "Poisson Multi-Bernoulli Mixture Conjugate Prior for Multiple Extended Target Filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 1, pp. 208–225, Feb. 2020, ISSN: 1557-9603. DOI: 10.1109/TAES.2019.2920220.

[26] G. Li, L. Kong, W. Yi, and X. Li, "Robust Poisson Multi-Bernoulli Mixture Filter With Unknown Detection Probability," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 1, pp. 886–899, Jan. 2021, ISSN: 1939-9359. DOI: 10.1109/TVT.2020.3047107.

[27] Á. F. García-Fernández, Y. Xia, and L. Svensson, "Poisson multi-Bernoulli mixture filter with general target-generated measurements and arbitrary clutter," *IEEE Transactions on Signal Processing*, vol. 71, pp. 1895–1906, 2023, ISSN: 1053-587X, 1941-0476. DOI: 10.1109/TSP.2023.3278944. arXiv: 2210.12983 [stat].

[28] X. Xie, Y. Wang, J. Guo, and R. Zhou, "The Multiple Model Poisson Multi-Bernoulli Mixture Filter for Extended Target Tracking," *IEEE Sensors Journal*, vol. 23, no. 13, pp. 14 304–14 314, Jul. 2023, ISSN: 1558-1748. DOI: 10.1109/JSEN.2023.3270272.

[29] M. Baerveldt, "Multiple Extended Object Tracking in Confined Waterways,"

[30] M. Fontana, Á. F. García-Fernández, and S. Maskell, *Poisson multi-Bernoulli mixture filter for trajectory measurements*, Apr. 2025. DOI: 10.48550/arXiv.2504.08421. arXiv: 2504.08421 [eess].

[31] Y. Xia, K. Granström, L. Svensson, Á. F. García-Fernández, and J. L. Williams, *Extended target Poisson multi-Bernoulli mixture trackers based on sets of trajectories*, Nov. 2019. DOI: 10.48550/arXiv.1911.09025. arXiv: 1911.09025 [eess].

[32] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, *Optuna: A Next-generation Hyperparameter Optimization Framework*, Jul. 2019. doi: 10.48550/arXiv.1907.10902. arXiv: 1907.10902 [cs].

[33] A. S. Rahmathullah, Á. F. García-Fernández, and L. Svensson, "Generalized optimal subpattern assignment metric," in *2017 20th International Conference on Information Fusion (Fusion)*, Jul. 2017, pp. 1–8. doi: 10.23919/ICIF.2017.8009645.

[34] K. Granström and U. Orguner, "On the reduction of Gaussian inverse Wishart mixtures," in *2012 15th International Conference on Information Fusion*, Jul. 2012, pp. 2162–2169.

[35] J. Ko, D. J. Klein, D. Fox, and D. Haehnel, "GP-UKF: Unscented kalman filters with Gaussian process prediction and observation models," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA: IEEE, Oct. 2007, pp. 1901–1907, isbn: 978-1-4244-0911-2. doi: 10.1109/IROS.2007.4399284.

# A

# PHD Filter Algorithm

[4] [13]

The PHD filter is a recursive Bayesian filter that propagates the first-order statistical moment (the PHD, or intensity) of the multi-target state, rather than the full multi-target probability density. This significantly reduces computational complexity. The integral of the PHD over a region gives the expected number of targets in that region.

## A.1. Definitions

- State Space: $\mathcal{X}$
- Measurement Space: $\mathcal{Z}$
- Time Step: $k$
- State Vector: $\mathbf{x} \in \mathcal{X}$
- Measurement Vector: $\mathbf{z} \in \mathcal{Z}$
- Measurement Set at time $k$: $Z_k = \{\mathbf{z}_1, \ldots, \mathbf{z}_{m_k}\}$
- Posterior Intensity at time $k-1$: $D_{k-1}(\mathbf{x})$
- Predicted Intensity at time $k$: $D_{k|k-1}(\mathbf{x})$
- Posterior Intensity at time $k$: $D_k(\mathbf{x})$
- Single-target transition density: $f_{k|k-1}(\mathbf{x}|\boldsymbol{\zeta})$ (probability density of state $\mathbf{x}$ at time $k$ given state $\boldsymbol{\zeta}$ at $k-1$)
- Probability of target survival: $p_{S,k}(\boldsymbol{\zeta})$
- Intensity of spawned targets: $\gamma_k^{\text{sp}}(\mathbf{x}|\boldsymbol{\zeta})$ (intensity of targets spawned from a target at state $\boldsymbol{\zeta}$)
- Intensity of new born targets: $\gamma_k(\mathbf{x})$
- Single-target measurement likelihood: $g_k(\mathbf{z}|\mathbf{x})$
- Probability of target detection: $p_{D,k}(\mathbf{x})$
- Clutter intensity (false alarms): $\kappa_k(\mathbf{z})$

## A.2. Algorithm: PHD Filter Recursion

**INPUT:**

- Posterior intensity from previous time step, $D_{k-1}(\mathbf{x})$.
- Measurement set at current time step, $Z_k$.

**OUTPUT:**

- Posterior intensity for current time step, $D_k(\mathbf{x})$.
- Estimated target state set, $\hat{X}_k$.

## STEP 1: PREDICTION

The predicted intensity $D_{k|k-1}(\mathbf{x})$ is the sum of the intensity from surviving targets and the intensity from spontaneously born targets.

**Intensity of Surviving Targets:** This component accounts for targets that existed at time $k-1$ and survived to time $k$. This includes both simple survival and spawning new targets.

$$D_{S,k|k-1}(\mathbf{x}) = \int \left[p_{S,k}(\boldsymbol{\zeta}) f_{k|k-1}(\mathbf{x}|\boldsymbol{\zeta}) + \gamma_k^{\mathsf{sp}}(\mathbf{x}|\boldsymbol{\zeta})\right] D_{k-1}(\boldsymbol{\zeta}) d\boldsymbol{\zeta} \tag{A.1}$$

**Intensity of New Born Targets:** This component accounts for new targets that appear at time $k$ independently of previous targets.

$$D_{B,k|k-1}(\mathbf{x}) = \gamma_k(\mathbf{x}) \tag{A.2}$$

**Total Predicted Intensity:** Combine the survival and birth intensities.

$$D_{k|k-1}(\mathbf{x}) = D_{S,k|k-1}(\mathbf{x}) + D_{B,k|k-1}(\mathbf{x}) \tag{A.3}$$

## STEP 2: UPDATE

The predicted intensity is updated using the measurement set $Z_k$. The posterior intensity consists of a term for missed detections and a sum of terms for each measurement in $Z_k$.

**Missed Detection Term:** This term represents the intensity of targets that were predicted to be present but were not detected.

$$D_{\mathsf{MD},k}(\mathbf{x}) = [1 - p_{D,k}(\mathbf{x})] D_{k|k-1}(\mathbf{x}) \tag{A.4}$$

**Measurement Update Term:** For each measurement $\mathbf{z} \in Z_k$, update the intensity. This term represents the possibility that the measurement $\mathbf{z}$ originated from a target.

$$\Psi_k(\mathbf{x}; \mathbf{z}) = \frac{p_{D,k}(\mathbf{x}) g_k(\mathbf{z}|\mathbf{x}) D_{k|k-1}(\mathbf{x})}{\kappa_k(\mathbf{z}) + \int p_{D,k}(\boldsymbol{\zeta}) g_k(\mathbf{z}|\boldsymbol{\zeta}) D_{k|k-1}(\boldsymbol{\zeta}) d\boldsymbol{\zeta}} \tag{A.5}$$

**Total Posterior Intensity:** The posterior intensity is the sum of the missed detection intensity and the updates from all measurements.

$$D_k(\mathbf{x}) = D_{\mathsf{MD},k}(\mathbf{x}) + \sum_{\mathbf{z} \in Z_k} \Psi_k(\mathbf{x}; \mathbf{z}) \tag{A.6}$$

## STEP 3: PRUNING & MERGING (For Gaussian Mixture Implementations)

When the PHD is implemented as a Gaussian Mixture (GM-PHD), the number of Gaussian components grows exponentially. To maintain computational tractability, this step is crucial.

- **Pruning:** Discard Gaussian components with weights below a predefined threshold $T_p$.
- **Merging:** Merge Gaussian components that are close to each other (e.g., their Mahalanobis distance is below a threshold $U_m$) into a single new Gaussian component.
- **Capping:** Limit the total number of Gaussian components to a maximum value $J_{\mathsf{max}}$.

### STEP 4: STATE EXTRACTION

The final step is to extract a discrete set of target state estimates from the posterior intensity $D_k(\mathbf{x})$.

**Estimate Number of Targets:** The expected number of targets $\hat{N}_k$ is the integral of the posterior intensity.

$$\hat{N}_k = \int D_k(\mathbf{x}) d\mathbf{x} \tag{A.7}$$

For a GM-PHD, this is the sum of the weights of all Gaussian components.

**Extract State Estimates:** Find the peaks (local maxima) of the posterior intensity function $D_k(\mathbf{x})$. The locations of the peaks with significant mass are taken as the state estimates.

For a GM-PHD, this typically involves selecting the means of the Gaussian components whose weights are above a certain threshold (e.g., $w > 0.5$).

$$\hat{X}_k = \{\hat{\mathbf{x}}_i \mid \hat{\mathbf{x}}_i \text{ is the mean of a component with significant weight}\} \tag{A.8}$$

# B

# Poisson Multi Bernoulli Mixture

## B.1. Poisson Multi-Bernoulli Mixture (PMBM) Filter

The PMBM filter is a sophisticated recursive Bayesian filter for multi-target tracking. It models the multi-target state as the union of two disjoint random finite sets: a Poisson Point Process (PPP) for targets that have not yet been detected, and a Multi-Bernoulli Mixture (MBM) for targets that have been detected at least once. This hybrid approach provides a more precise posterior density compared to the PHD filter by maintaining individual track hypotheses and their data association uncertainties.

### B.1.1. Definitions

- State Space, Measurement Space, Time Step, etc.: Same as PHD Filter.
- **Undetected Target Intensity (PPP):** The intensity of the Poisson Point Process, $D_{k-1}^{u}(\mathbf{x})$, represents the expected density of targets that have not yet been detected.
- **Detected Target Set (MBM):** The set of detected targets is represented by a Multi-Bernoulli Mixture. An MBM is a weighted sum of multi-Bernoulli (MB) components.
- **Global Hypothesis, $h$:** A specific data association history, which maps measurements to tracks over time. The full posterior is a mixture of these global hypotheses.
- **Global Hypothesis Weight, $w^{(h)}$:** The probability of a specific global hypothesis $h$.
- **Track Hypothesis (Bernoulli Component):** For each detected target track $i$ within a global hypothesis $h$, its state is modeled as a Bernoulli RFS with:
  - **Existence Probability:** $r^{(h,i)}$, the probability that track $i$ exists.
  - **State PDF:** $p^{(h,i)}(\mathbf{x})$, the probability density function of the state of track $i$.
- Single-target transition density, survival probability, etc.: Same as PHD Filter.

### B.1.2. Algorithm: PMBM Filter Recursion

**INPUT:**

- A set of weighted global hypotheses from the previous time step, $\{w_{k-1}^{(h)}, \{r_{k-1}^{(h,i)}, p_{k-1}^{(h,i)}(\mathbf{x})\}_{i=1}^{N_{k-1}^{(h)}}\}_{h=1}^{H_{k-1}}$.
- The intensity of the undetected targets PPP, $D_{k-1}^{u}(\mathbf{x})$.
- Measurement set at current time step, $Z_k$.

**OUTPUT:**

- Updated set of weighted global hypotheses, $\{w_{k}^{(h)}, \{r_{k}^{(h,i)}, p_{k}^{(h,i)}(\mathbf{x})\}_{i=1}^{N_{k}^{(h)}}\}_{h=1}^{H_{k}}\}$.
- Updated intensity of the undetected targets PPP, $D_{k}^{u}(\mathbf{x})$.
- Estimated target state set, $\hat{X}_k$.

STEP 1: PREDICTION

The prediction step is performed independently for the undetected and detected target components.

**Predict Undetected Targets (PPP):** The PPP intensity is predicted forward in time, accounting for surviving undetected targets and newly born targets.

$$D^u_{k|k-1}(\mathbf{x}) = \gamma_k(\mathbf{x}) + \int p_{S,k}(\boldsymbol{\zeta}) f_{k|k-1}(\mathbf{x}|\boldsymbol{\zeta}) D^u_{k-1}(\boldsymbol{\zeta}) d\boldsymbol{\zeta} \tag{B.1}$$

**Predict Detected Targets (MBM):** For each global hypothesis $h$ and each track $i$ within it, predict the state PDF and update the existence probability.

$$r^{(h,i)}_{k|k-1} = r^{(h,i)}_{k-1} \int p_{S,k}(\boldsymbol{\zeta}) p^{(h,i)}_{k-1}(\boldsymbol{\zeta}) d\boldsymbol{\zeta} \tag{B.2}$$

$$p^{(h,i)}_{k|k-1}(\mathbf{x}) = \frac{\int p_{S,k}(\boldsymbol{\zeta}) f_{k|k-1}(\mathbf{x}|\boldsymbol{\zeta}) p^{(h,i)}_{k-1}(\boldsymbol{\zeta}) d\boldsymbol{\zeta}}{\int p_{S,k}(\boldsymbol{\zeta}) p^{(h,i)}_{k-1}(\boldsymbol{\zeta}) d\boldsymbol{\zeta}} \tag{B.3}$$

STEP 2: UPDATE

The update step is more complex, involving the creation of new hypotheses based on measurement associations.

**Update Undetected Targets (PPP):** The intensity of undetected targets is updated to account for the possibility that they were not detected at time $k$.

$$D^u_k(\mathbf{x}) = [1 - p_{D,k}(\mathbf{x})] D^u_{k|k-1}(\mathbf{x}) \tag{B.4}$$

**Create Single-Target Hypotheses:** For each measurement $\mathbf{z} \in Z_k$, create a set of new track hypotheses. Each new hypothesis corresponds to the possibility that measurement $\mathbf{z}$ was generated by:

- A previously detected target $i$ from a prior global hypothesis $h$.
- A previously undetected target (a "new" target from the PPP).
- Clutter.

**Construct New Global Hypotheses:** Create a new set of global hypotheses by considering all possible valid combinations of single-target hypotheses (i.e., all valid partitions of the measurement set $Z_k$). Each new global hypothesis $h'$ corresponds to a unique assignment of measurements to tracks (new or existing) or clutter.

**Update Detected Tracks and Compute Hypothesis Weights:** For each new global hypothesis $h'$:

- For detected tracks associated with measurement $\mathbf{z}$: Update the Bernoulli parameters $(r, p)$ using the standard Kalman-like update for the state PDF $p(\mathbf{x})$ and a likelihood-based update for the existence probability $r$.
- For new tracks created from measurement $\mathbf{z}$: Initialize a new Bernoulli component. The existence probability is calculated based on the ratio of the new target likelihood to the clutter likelihood. The state PDF is initialized based on the measurement $\mathbf{z}$.
- For tracks (new or old) that were not detected (missed detection): Update their existence probability by multiplying with $(1 - p_D)$.
- Compute the weight $w^{(h')}_k$ for the new global hypothesis. This weight is proportional to the prior weight and the likelihood of the specific data association defined by $h'$.

STEP 3: HYPOTHESIS MANAGEMENT

The number of global hypotheses grows exponentially. Pruning and merging are essential to keep the filter tractable.

- **Pruning:** Discard global hypotheses whose weights $w_k^{(h')}$ fall below a threshold.
- **Merging:** Merge global hypotheses that are similar (e.g., have similar track sets and data associations) into a single representative hypothesis.
- **Capping:** Limit the total number of global hypotheses to a maximum value $H_{\text{max}}$.
- **Recycling:** Components of pruned hypotheses corresponding to high-probability tracks can be "recycled" back into the PPP of undetected targets to prevent track loss.

STEP 4: STATE EXTRACTION

Extract a discrete set of target states from the updated PMBM density.

- **Select Best Hypothesis:** Find the global hypothesis $\hat{h}$ with the maximum weight, $w_k^{(\hat{h})}$.
- **Extract State Estimates:** Within the best hypothesis $\hat{h}$, extract the state estimates for all tracks whose existence probability $r_k^{(\hat{h},i)}$ is above a certain threshold (e.g., $r > 0.5$). The state estimate is typically the mean of the corresponding state PDF $p_k^{(\hat{h},i)}(\mathbf{x})$.

$$\hat{X}_k = \{\hat{\mathbf{x}}_i \mid \hat{\mathbf{x}}_i = \mathbb{E}[p_k^{(\hat{h},i)}(\mathbf{x})] \text{ and } r_k^{(\hat{h},i)} > 0.5\} \tag{B.5}$$

# C

# Gamma Gaussian Inverse Wishart Component

Gamma Gaussian inverse Wishart (GGIW) components are used to model the distribution of the state of a target in a Bayesian filtering context, particularly in the context of tracking extended targets. The GGIW component is a conjugate prior for the parameters of a Gaussian distribution, which allows for efficient updates and predictions. It consists of three main elements:

- An Gamma distribution for the target's expected number of measurements.
- A Gaussian distribution for the target's kinematic state.
- An Inverse Wishart distribution as a prior for the target's size or extent.

The target extent is as a random matrix that describes the shape and size of the target.

The implementation presented in this thesis is derived form Granstroms implementation in [13].

the extended target state is given by

$$\xi_k \triangleq (\gamma_k, x_k, X_k)$$

The extended target state $\xi_k$ at time $k$ consists of the following components:

- $\gamma_k$: A random variable representing the measurement rate $(\gamma_k > 0)$.
- $x_k$: The kinematic state vector, defined as $x_k = [p_k, v_k, \omega_k]^T \in \mathbb{R}^5$, where:
  - $p_k \in \mathbb{R}^2$: The target's position.
  - $v_k \in \mathbb{R}^2$: The target's velocity.
  - $\omega_k \in \mathbb{R}^1$: The target's turn rate.
- $X_k$: A random matrix representing the extension state, $X_k \in \mathbb{S}^2_{++}$ (a symmetric positive definite $d \times d$ matrix).
- $d$: The dimension of the matrix $X_k$ (here $d = 2$).

$$p(\xi_k|Z^k) = p(\gamma_k|Z^k)p(x_k|Z^k)p(X_k|Z^k) = \mathcal{G}(\gamma_k; \alpha_{k|k}, \beta_{k|k})\mathcal{N}(x_k; m_{k|k}, P_{k|k}) \times \mathcal{IW}_d(X_k; v_{k|k}, V_{k|k})$$

- $p(\xi_k|Z^k)$: Posterior probability density of the target state $\xi_k$ given measurement history $Z^k$.
- $\mathcal{G}(\gamma_k; \alpha_{k|k}, \beta_{k|k})$: Gamma distribution for the measurement rate $\gamma_k$ with shape $\alpha_{k|k}$ and inverse scale $\beta_{k|k}$.
- $\mathcal{N}(x_k; m_{k|k}, P_{k|k})$: Gaussian distribution for the kinematic state $x_k$ with mean $m_{k|k}$ and covariance $P_{k|k}$.

- $\mathcal{IW}_d(X_k; v_{k|k}, V_{k|k})$: Inverse Wishart distribution for the extension state $X_k$ with degrees of freedom $v_{k|k}$ and parameter matrix $V_{k|k}$ ($d = 2$).

## C.1. prediction

Gamma distribution prediction:

$$\beta^{(j)}_{k+1|k} = \frac{\beta^{(j)}_{k|k}}{\eta_k}, \alpha k + 1|k^{(j)} = \frac{\alpha k|k^{(j)}}{\eta_k} \tag{C.1}$$

- $\eta_k$: forgetting factor to increase variance of $\gamma$ estimate

Kinematic state prediction:

$$m^{(j)}_{k+1|k} = f(m^{(j)}_{k|k})$$

$$P^{(j)}_{k+1|k} = F^{(j)}_{k|k} P^{(j)}_{k|k} (F^{(j)}_{k|k})^T + Q$$

$F^{(j)}_{k|k}$: Jacobian of $f(x)$ w.r.t. $x$ evaluated at $m^{(j)}_{k|k}$

Inverse Wishart distribution prediction:

$$v^{(j)}_{k+1|k} = 2d + 2 + e^{-T_s/\tau}(v^{(j)}_{k|k} - 2d - 2)$$

$$V^{(j)}_{k+1|k} = (v^{(j)}_{k+1|k} - 2d - 2)(v^{(j)}_{k|k} - 2d - 2)^{-1} V^{(j)}_{k|k}$$

$\tau$: temporal decay constant for degrees of freedom $v$

## C.2. Measurement Update

### C.2.1. missed detection

### C.2.2. detection

$W$: set of measurements associated with target $j$ at time $k$.

Gamma Update:

$$\alpha^{(j,W)}_{k|k} = \alpha^{(j)}_{k|k-1} + |W|, \quad \beta^{(j,W)}_{k|k} = \beta^{(j)}_{k|k-1} + 1$$

kinematic Update:

$$m^{(j,W)}_{k|k} = m^{(j)}_{k|k-1} + K^{(j,W)}_{k|k-1} \epsilon^{(j,W)}_{k|k-1}$$

$$P^{(j,W)}_{k|k} = P^{(j)}_{k|k-1} - K^{(j,W)}_{k|k-1} H_k P^{(j)}_{k|k-1}$$

Inverse Wishart Update:

$$v^{(j,W)}_{k|k} = v^{(j)}_{k|k-1} + |W|$$

$$V^{(j,W)}_{k|k} = V^{(j)}_{k|k-1} + \hat{N}^{(j,W)}_{k|k-1} + \hat{Z}^{(j,W)}_{k|k-1}$$

Update factors and cross covariance calulation:

$$\hat{X}^{(j)}_{k|k-1} = V^{(j)}_{k|k-1}(v^{(j)}_{k|k-1} - 2d - 2)^{-1}$$

measurement covariance, due to the target extent and measurement noise covariance.

$$\hat{R}^{(j,W)}_{k|k-1} = \rho \hat{X}^{(j)}_{k|k-1} + R(H_k m^{(j)}_{k|k-1})$$

Measurement residual

$$\epsilon^{(j,W)}_{k|k-1} = \bar{\mathbf{z}}^{W}_{k} - H_k m^{(j)}_{k|k-1}$$

$$N^{(j,W)}_{k|k-1} = \epsilon^{(j,W)}_{k|k-1}(\epsilon^{(j,W)}_{k|k-1})^T$$

$$S^{(j,W)}_{k|k-1} = H_k P^{(j)}_{k|k-1} H^T_k + \frac{\hat{R}^{(j,W)}_{k|k-1}}{|W|}$$

$$K^{(j,W)}_{k|k-1} = P^{(j)}_{k|k-1} H^T_k (S^{(j,W)}_{k|k-1})^{-1}$$

$$\hat{Z}^{(j,W)}_{k|k-1} = (\hat{X}^{(j)}_{k|k-1})^{1/2}(\hat{R}^{(j,W)}_{k|k-1})^{-1/2} Z^W_k (\hat{R}^{(j,W)}_{k|k-1})^{-T/2}(\hat{X}^{(j)}_{k|k-1})^{T/2}$$

$$\hat{N}^{(j,W)}_{k|k-1} = (\hat{X}^{(j)}_{k|k-1})^{1/2}(S^{(j,W)}_{k|k-1})^{-1/2} N^{(j,W)}_{k|k-1}(S^{(j,W)}_{k|k-1})^{-T/2}(\hat{X}^{(j)}_{k|k-1})^{T/2}$$

And finally the likelyhood of the measurement set $Z_k$ given the target state $\xi_k$ is given by:

$$\mathcal{L}^{(j,W)}_k = \frac{(2\pi)^{-|W|n_z/2}|\hat{X}^{(j)}_{k|k-1}|^{|W|/2}}{|S^{(j,W)}_{k|k-1}|^{1/2}|\hat{R}^{(j,W)}_{k|k-1}|^{1/2}} \times \frac{|V^{(j)}_{k|k-1}|^{\frac{v^{(j)}_{k|k-1}-d-1}{2}}\Gamma_d(\frac{v^{(j,W)}_{k|k}}{2})}{|V^{(j,W)}_{k|k}|^{\frac{v^{(j,W)}_{k|k}-d-1}{2}}\Gamma_d(\frac{v^{(j)}_{k|k-1}}{2})} \times \frac{\Gamma(\alpha^{(j,W)}_{k|k})(\beta^{(j)}_{k|k-1})^{\alpha^{(j)}_{k|k-1}}}{\Gamma(\alpha^{(j)}_{k|k-1})(\beta^{(j,W)}_{k|k})^{\alpha^{(j,W)}_{k|k}}}$$

- $n_z$: dimension of the measurement vector $z^{(j)}_k$ (here $n_z = 2$) - $d$: dimension of the extension matrix $X_k$ (here $d = 2$) - Note: The exponents in the paper image for $\det(V)$ terms are $(v - d - 1)/2$. Other literature sometimes uses $v/2$.

## C.3. Mixture Reduction

Mixture reduction

[34]

# D

# Star-Convex Gaussian Process Filter

This chapter details the implementation of a filter for an extended target whose shape is modelled as a star-convex object using a Gaussian Process (GP). The core of this method is the formulation of the GP regression as a recursive state-space model, which is then augmented with the target's kinematic state. This allows for joint estimation of both shape and kinematics within a Bayesian filtering framework. We describe the background, the state-space model, and the specific time and measurement update steps for both the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) implementations.

This work is fully based on [15], which provides a comprehensive framework for tracking extended targets using Gaussian Processes. The implementation presented here is adapted to the specific requirements of star-convex shape modelling and recursive estimation. The unscented Kalman filter formulation is derived from Ko et al. [35] and Wan and Van Der Merwe [9].

## D.1. Background and Definitions

*Star-Convex Shape Model*
A target's shape is described as a star-convex set, where the boundary can be represented in a local polar coordinate system by a radial function $r = f(\theta)$. This function maps an angle $\theta$ to a radial distance from the target's centre. Measurements are assumed to originate from this boundary or the surface it encloses.

*Gaussian Process for the Radial Function*
The unknown radial function $f(\theta)$ is modelled using a Gaussian Process (GP), which provides a flexible, non-parametric prior over functions. A GP is fully specified by a mean function $\mu(\theta)$ and a covariance function $k(\theta, \theta')$. For a target contour, the radial function must be periodic with a period of $2\pi$. This is encoded into the covariance function. A commonly used choice is the periodic squared exponential kernel, which also includes a term for the mean radius variance:

$$k_{tot}(\theta, \theta') = \sigma_f^2 \exp\left(-\frac{2\sin^2\left(\frac{|\theta-\theta'|}{2}\right)}{l^2}\right) + \sigma_r^2 \tag{D.1}$$

where $\sigma_f^2$ is the signal variance, $l$ is the length scale, and $\sigma_r^2$ is the prior variance of the mean radius.

*Recursive GP State-Space Formulation*
Standard GP regression is a batch process, making it unsuitable for online tracking. To enable recursive estimation, the GP is approximated using a finite set of $M$ basis points (test inputs $\mathbf{u}^f$). The corresponding function values $\mathbf{f}$ at these points are treated as the state of the GP. This transforms the GP regression problem into a linear state-space estimation problem that can be solved with a Kalman filter.

The likelihood of a measurement $z_k$ at input $u_k$ given the state $\mathbf{f}$ is:

$$p(z_k|\mathbf{f}) = \mathcal{N}\left(z_k; H^{\mathbf{f}}(u_k)\mathbf{f}, R^{\mathbf{f}}(u_k)\right) \tag{D.2}$$

where

$$H^{\mathbf{f}}(u_k) = K(u_k, \mathbf{u}^{\mathbf{f}})[K(\mathbf{u}^{\mathbf{f}}, \mathbf{u}^{\mathbf{f}})]^{-1} \tag{D.3}$$

$$R^{\mathbf{f}}(u_k) = k(u_k, u_k) + R - H^{\mathbf{f}}(u_k)K(\mathbf{u}^{\mathbf{f}}, u_k) \tag{D.4}$$

*Augmented State Vector*

The complete system state $\mathbf{x}_k$ is an augmentation of the target's kinematic state $\mathbf{x}_{ki,k}$ and its extent state $\mathbf{x}_{e,k}$ (the GP state $\mathbf{f}$).

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_{ki,k} \\ \mathbf{x}_{e,k} \end{bmatrix} \tag{D.5}$$

*Augmented State Covariance Matrix*

The covariance matrix $\mathbf{P}_k$ for the augmented state $\mathbf{x}_k$ is block-diagonal, reflecting the independence between the kinematic and extent states:

$$\mathbf{P}_k = \begin{bmatrix} \mathbf{P}_{ki,k} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{e,k} \end{bmatrix} \tag{D.6}$$

where:

- $\mathbf{P}_{ki,k}$ is the covariance matrix for the kinematic state $\mathbf{x}_{ki,k}$.
- $\mathbf{P}_{e,k}$ is the covariance matrix for the extent state $\mathbf{x}_{e,k}$, derived from the GP model.

The kinematic state $\mathbf{x}_{ki,k}$ typically includes position, orientation, and their velocities, i.e., $\mathbf{x}_{ki,k} = [(\mathbf{x}_{c,k})^T, x_{\phi,k}, (\mathbf{x}_{*,k})^T]^T$.

## D.2. Time Update (Prediction)

The time update, or prediction step, propagates the state mean and covariance forward in time using a linear dynamic model. This step is identical for both EKF and UKF implementations.

*Dynamic Model*

The state evolves according to the linear stochastic equation:

$$\mathbf{x}_{k+1} = \mathbf{F}_k\mathbf{x}_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k) \tag{D.7}$$

The state transition matrix $\mathbf{F}_k$ and process noise covariance $\mathbf{Q}_k$ are block-diagonal, decoupling the kinematic and extent dynamics.

$$\mathbf{F}_k = \begin{bmatrix} \mathbf{F}_{ki,k} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{e,k} \end{bmatrix}, \quad \mathbf{Q}_k = \begin{bmatrix} \mathbf{Q}_{ki,k} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{e,k} \end{bmatrix} \tag{D.8}$$

The kinematic components ($\mathbf{F}_{ki,k}$, $\mathbf{Q}_{ki,k}$) are derived from a standard motion model, such as Constant Velocity or Constant Turn. The extent dynamics incorporate a forgetting factor $\alpha$ to allow the shape to adapt over time:

$$\mathbf{F}_{e,k} = e^{-\alpha T}\mathbf{I}_M \tag{D.9}$$

$$\mathbf{Q}_{e,k} = (1 - e^{-2\alpha T})\mathbf{k}(\theta, \theta) \tag{D.10}$$

*Prediction Equations*

Given the posterior estimate $\{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}\}$ from the previous step, the predicted state mean and covariance are computed as:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1|k-1} \tag{D.11}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \tag{D.12}$$

## D.3. Measurement Update (Correction)

The measurement update incorporates a new measurement $\mathbf{y}_k$ to correct the predicted state. This step is nonlinear and is where the EKF and UKF implementations differ.

*Measurement Model*

A single measurement $\mathbf{y}_k^{(i)}$ is related to the state by the nonlinear function $h(\cdot)$:

$$\mathbf{y}_k^{(i)} = h(\mathbf{x}_k, \mathbf{e}_k^{(i)}) = \mathbf{x}_{c,k} + s_k^{(i)} \mathbf{p}_k^{(i)} H_{p,k}^{(i)} \mathbf{x}_{e,k} + \text{noise terms} \tag{D.13}$$

where $\mathbf{x}_{c,k}$ and $\mathbf{x}_{e,k}$ are sub-states of $\mathbf{x}_k$. The total measurement noise covariance $\mathbf{R}_{tr,k}^{(i)}$ is the sum of the sensor noise $\mathbf{R}_k$ and the GP regression uncertainty $\mathbf{R}_{GP,k}^{(i)}$.

$$\mathbf{R}_{tr,k}^{(i)} = \mathbf{R}_{GP,k}^{(i)} + \mathbf{R}_k \tag{D.14}$$

where $\mathbf{R}_{GP,k}^{(i)} = (s_k^{(i)} \mathbf{p}_k^{(i)}) R_{p,k}^{(i)} (s_k^{(i)} \mathbf{p}_k^{(i)})^T$.

### D.3.1. EKF Measurement Update

The EKF linearizes the measurement function $h(\mathbf{x}_k)$ by computing its Jacobian matrix $\mathbf{H}_k$ at the predicted state $\hat{\mathbf{x}}_{k|k-1}$. The update then proceeds with the standard linear Kalman filter equations.

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_{tr,k} \tag{D.15}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \tag{D.16}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{y}_k - h(\hat{\mathbf{x}}_{k|k-1})) \tag{D.17}$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \tag{D.18}$$

The expressions for the Jacobian $\mathbf{H}_k$ are complex and require differentiating through the GP model and coordinate transformations.

### D.3.2. UKF Measurement Update

The UKF avoids computing Jacobians by propagating a set of deterministically chosen sigma points through the non-linear measurement function.

1. **Generate Sigma Points:** Calculate $2L+1$ sigma points $\boldsymbol{\chi}_{k|k-1}$ from the predicted mean $\hat{\mathbf{x}}_{k|k-1}$ and covariance $\mathbf{P}_{k|k-1}$, where $L$ is the dimension of the state $\mathbf{x}_k$.

$$\boldsymbol{\chi}_{k|k-1} = \left[ \hat{\mathbf{x}}_{k|k-1} \quad \hat{\mathbf{x}}_{k|k-1} \pm \sqrt{(L+\lambda)\mathbf{P}_{k|k-1}} \right] \tag{D.19}$$

2. **Propagate Sigma Points:** Pass each sigma point $\boldsymbol{\chi}_j$ through the non-linear measurement function $h(\cdot)$ to generate a set of transformed points $\boldsymbol{\mathcal{Y}}_j$.

$$\boldsymbol{\mathcal{Y}}_{j,k|k-1} = h(\boldsymbol{\chi}_{j,k|k-1}) \tag{D.20}$$

3. **Calculate Predicted Measurement Statistics:** Compute the weighted mean and covariances of the transformed points.

$$\hat{\mathbf{y}}_k^- = \sum_{j=0}^{2L} W_j^{(m)} \boldsymbol{\mathcal{Y}}_{j,k|k-1} \tag{D.21}$$

$$\mathbf{P}_{yy} = \sum_{j=0}^{2L} W_j^{(c)} (\boldsymbol{\mathcal{Y}}_{j,k|k-1} - \hat{\mathbf{y}}_k^-)(\boldsymbol{\mathcal{Y}}_{j,k|k-1} - \hat{\mathbf{y}}_k^-)^T + \mathbf{R}_{tr,k} \tag{D.22}$$

$$\mathbf{P}_{xy} = \sum_{j=0}^{2L} W_j^{(c)} (\boldsymbol{\chi}_{j,k|k-1} - \hat{\mathbf{x}}_{k|k-1})(\boldsymbol{\mathcal{Y}}_{j,k|k-1} - \hat{\mathbf{y}}_k^-)^T \tag{D.23}$$

4. **Update State Estimate:** Compute the Kalman gain and update the state mean and covariance.

$$\mathbf{K}_k = \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1} \tag{D.24}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k^-) \tag{D.25}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{P}_{yy}\mathbf{K}_k^T \tag{D.26}$$

This procedure provides a more accurate approximation of the posterior mean and covariance than the EKF, especially for highly nonlinear systems, without requiring analytical differentiation.

# E

# Elliptical Principal Axes Model

This chapter details the implementation of a filter for an extended target whose shape is modelled as an ellipse. The core of this method, termed the Principal Axes Kalman Filter (PAKF), is the direct estimation of the ellipse's extent parameters (semi-axes lengths and orientation) from a set of sensor measurements. This is achieved by computing the principal axes of the measurement point cloud's dispersion matrix via eigen-decomposition. The target's kinematic and extent states are decoupled and estimated separately within a standard Bayesian filtering framework.

This work is fully based on [16], which provides a computationally efficient framework for tracking elliptical extended targets, particularly when a high number of measurements is available.

## E.1. Background and Definitions

*Elliptical Shape Model*

A target's shape is approximated by a two-dimensional ellipse. The extent is explicitly parametrised by the length of the major semi-axis $a$, the length of the minor semi-axis $b$, and the orientation angle $\theta$. Measurements are assumed to originate from the area enclosed by this ellipse.

*Decoupled State Vector*

The complete system state is separated into a kinematic state $\mathbf{x}_k$ and an extent (or parameter) state $\mathbf{p}_k$. This decoupling allows for independent prediction and update steps for motion and shape.

- The kinematic state $\mathbf{x}_k$ contains the target's Cartesian centre position and corresponding velocity vector:
$$\mathbf{x}_k = \begin{bmatrix} \mathbf{m}_k \\ \dot{\mathbf{m}}_k \end{bmatrix}.$$
- The extent state $\mathbf{p}_k$ explicitly contains the orientation and the lengths of the semi-axes:
$$\mathbf{p}_k = \begin{bmatrix} \theta_k \\ a_k \\ b_k \end{bmatrix}.$$

*Principal Axes from Eigen-decomposition*

The parameters for the extent measurement vector are derived from the dispersion (scatter) matrix $\Sigma_D$ of the measurement point cloud $\mathcal{Y}_k = \{y_k^i\}_{i=1}^{n_k}$. The dispersion matrix captures the spatial spread of the measurements around their mean $\bar{y}_k$:

$$\bar{y}_k = \frac{1}{n_k}\sum_{i=1}^{n_k} y_k^i, \qquad \Sigma_D = \frac{1}{n_k - 1}\sum_{i=1}^{n_k}\big(y_k^i - \bar{y}_k\big)\big(y_k^i - \bar{y}_k\big)^\top,$$

where the factor $1/(n_k - 1)$ gives the unbiased sample covariance (use $1/n_k$ for the ML estimator if preferred).

To account for sensor noise, the dispersion matrix is corrected by subtracting the measurement noise covariance $C_\vartheta^x$:

$$\widehat{\Sigma}_D = \Sigma_D - C_\vartheta^x.$$

An eigen-decomposition of the corrected matrix, $\widehat{\Sigma}_D = V\Lambda V^\top$, yields eigenvalues $\lambda_1, \lambda_2$ and corresponding eigenvectors. Let $\lambda_1 \geq \lambda_2$ and let $v_1 = (v_{1,x}, v_{1,y})^\top$ be the eigenvector associated with $\lambda_1$. The ellipse parameters are then extracted as:

$$a_k = 2\sqrt{\lambda_1}, \tag{E.1}$$

$$b_k = 2\sqrt{\lambda_2}, \tag{E.2}$$

$$\theta_k = \operatorname{atan2}\big(v_{1,y},\, v_{1,x}\big), \tag{E.3}$$

and the extent measurement vector is $\mathbf{z}_k^p = [\theta_k,\, a_k,\, b_k]^\top$.

## E.2. Time Update (Prediction)

The time update propagates the kinematic and extent state estimates forward in time using a linear dynamic model.

*Dynamic Model*

The state evolves according to a linear stochastic model. For the kinematic state:

$$\mathbf{x}_{k|k-1} = A_x\, \mathbf{x}_{k-1|k-1} + w_k^x, \qquad w_k^x \sim \mathcal{N}\big(0,\, C_\omega^x\big).$$

For the extent state:

$$\mathbf{p}_{k|k-1} = A_p\, \mathbf{p}_{k-1|k-1} + w_k^p, \qquad w_k^p \sim \mathcal{N}\big(0,\, C_\omega^p\big).$$

Typically, a nearly constant velocity model is used for the kinematics and a random walk model for the extent (e.g. $A_p = I_3$). The process noise covariances $C_\omega^x$ and $C_\omega^p$ are set according to expected target dynamics.

*Prediction Equations*

Given the posterior estimates from the previous step, the predicted state means and covariances are:

$$\hat{\mathbf{x}}_{k|k-1} = A_x\, \hat{\mathbf{x}}_{k-1|k-1}, \tag{E.4}$$

$$C_{k|k-1}^x = A_x\, C_{k-1|k-1}^x\, A_x^\top + C_\omega^x, \tag{E.5}$$

$$\hat{\mathbf{p}}_{k|k-1} = A_p\, \hat{\mathbf{p}}_{k-1|k-1}, \tag{E.6}$$

$$C_{k|k-1}^p = A_p\, C_{k-1|k-1}^p\, A_p^\top + C_\omega^p. \tag{E.7}$$

## E.3. Measurement Update (Correction)

The measurement update incorporates new measurements to correct the predicted state. The raw point cloud is first processed to form two distinct measurement vectors.

*Measurement Model*

From the set of $n_k$ raw measurements $\{y_k^i\}$, two processed measurements are formed:

- The kinematic measurement $\mathbf{z}_k^x$ is the sample mean of the points:

$$\mathbf{z}_k^x = \bar{y}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} y_k^i.$$

- The extent measurement $\mathbf{z}_k^p$ is derived from the eigen-decomposition of the (corrected) dispersion matrix:

$$\mathbf{z}_k^p = \begin{bmatrix} \theta_k \\ a_k \\ b_k \end{bmatrix}.$$

Both resulting measurement models are written as linear Gaussian models:

$$\mathbf{z}_k^x = H_x\,\mathbf{x}_k + v_k^x, \qquad v_k^x \sim \mathcal{N}\big(0,\, C_\vartheta^x\big),$$

$$\mathbf{z}_k^p = H_p\,\mathbf{p}_k + v_k^p, \qquad v_k^p \sim \mathcal{N}\big(0,\, C_\vartheta^p\big),$$

where $H_x = [I_2\ 0_{2\times2}]$ and $H_p = I_3$.

*Kalman Filter Update Equations*

Since the measurement models are linear, the standard Kalman filter update equations are applied independently to the kinematic and extent states. For a generic state $\mathbf{s}$ (where $\mathbf{s}$ can be $\mathbf{x}$ or $\mathbf{p}$) with measurement matrix $H$ and measurement noise covariance $C_\vartheta$:

$$S_k = H\,C_{k|k-1}\,H^\top + C_\vartheta, \tag{E.8}$$

$$K_k = C_{k|k-1}\,H^\top\,S_k^{-1}, \tag{E.9}$$

$$\hat{\mathbf{s}}_{k|k} = \hat{\mathbf{s}}_{k|k-1} + K_k\big(\mathbf{z}_k - H\,\hat{\mathbf{s}}_{k|k-1}\big), \tag{E.10}$$

$$C_{k|k} = \big(I - K_k H\big)\,C_{k|k-1}. \tag{E.11}$$

This two-step update process—first for kinematics, then for extent—completes the estimation cycle.

# F

# Motion Models

## F.1. Constant Velocity

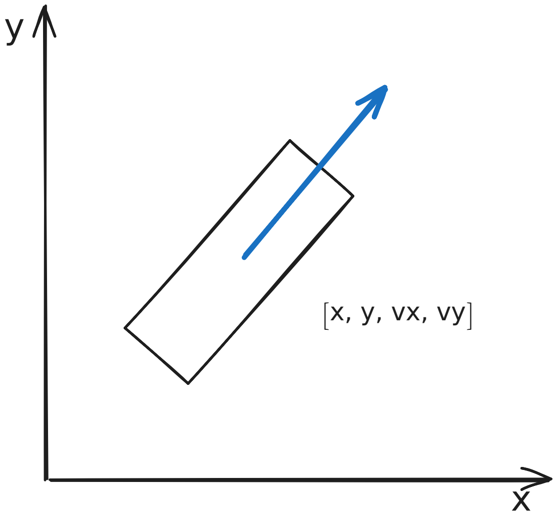


**Figure F.1:** Constant Velocity Motion Model

$$\mathbf{x_k} = Fx_{k-1} + w_k \tag{F.1}$$

### F.1.1. Transition Model

The transition model for the constant velocity motion model is given by:

$$\begin{bmatrix} p_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} p_k + v_k \cdot T_s \\ v_k \end{bmatrix} \tag{F.2}$$

where:

$p_k$ is the position at time $k$,<br>
$v_k$ is the velocity at time $k$,<br>
$T_s$ is the sampling time.

$$F = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{F.3}$$

### F.1.2. Noise Model

the noise is modelled as zero mean gaussian noise with covariance $\sigma_a$ on the acceleration:

$$w_k \sim \mathcal{N}(0, Q) \tag{F.4}$$

$$G = \begin{bmatrix} \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \tag{F.5}$$

$$Q = GG^T\sigma_a^2 = \begin{bmatrix} \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} & 0 \\ 0 & \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & 0 & \Delta t^2 & 0 \\ 0 & \frac{\Delta t^3}{2} & 0 & \Delta t^2 \end{bmatrix} \sigma_a^2 \tag{F.6}$$

## F.2. Constant Acceleration

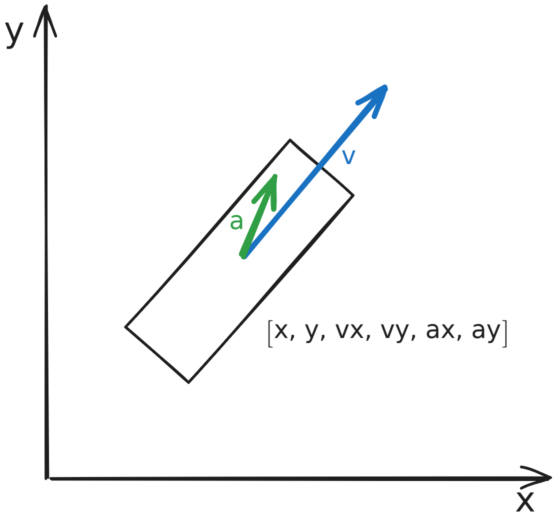


**Figure F.2:** Constant Acceleration Motion Model

$$\mathbf{x_k} = Fx_{k-1} + w_k \tag{F.7}$$

### F.2.1. Transition Model

The transition model for the constant acceleration motion model is given by:

$$\begin{bmatrix} p_{k+1} \\ v_{k+1} \\ a_{k+1} \end{bmatrix} = \begin{bmatrix} p_k + v_k \cdot T_s + \frac{1}{2}a_k \cdot T_s^2 \\ v_k + a_k \cdot T_s \\ a_k \end{bmatrix} \tag{F.8}$$

where:

$p_k$ is the position at time $k$,
$v_k$ is the velocity at time $k$,
$a_k$ is the acceleration at time $k$,
$T_s$ is the sampling time.

$$F = \begin{bmatrix} 1 & 0 & \Delta t & 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & 1 & 0 & \Delta t & 0 & \frac{\Delta t^2}{2} \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{F.9}$$

### F.2.2. Noise Model

The noise is modelled as zero-mean Gaussian noise with covariance $\sigma_j$ on the jerk:

$$w_k \sim \mathcal{N}(0, Q) \tag{F.10}$$

$$G = \begin{bmatrix} \frac{\Delta t^3}{6} & 0 \\ 0 & \frac{\Delta t^3}{6} \\ \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \tag{F.11}$$

$$Q = GG^T\sigma_a^2 = \begin{bmatrix} \frac{\Delta t^6}{36} & 0 & \frac{\Delta t^5}{12} & 0 & \frac{\Delta t^4}{6} & 0 \\ 0 & \frac{\Delta t^6}{36} & 0 & \frac{\Delta t^5}{12} & 0 & \frac{\Delta t^4}{6} \\ \frac{\Delta t^5}{12} & 0 & \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} & 0 \\ 0 & \frac{\Delta t^5}{12} & 0 & \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} \\ \frac{\Delta t^4}{6} & 0 & \frac{\Delta t^3}{2} & 0 & \Delta t^2 & 0 \\ 0 & \frac{\Delta t^4}{6} & 0 & \frac{\Delta t^3}{2} & 0 & \Delta t^2 \end{bmatrix} \sigma_a^2 \tag{F.12}$$

## F.3. Constant Turn Rate and Velocity

### F.3.1. Transition Model

The continuous-time transition model for the constant turn rate and velocity motion model is given by:

$$f_c trv = \begin{bmatrix} v\cos(\psi) \\ v\sin(\psi) \\ \dot{\psi} \\ 0 \\ 0 \end{bmatrix} \tag{F.13}$$

The discrete-time transition functions are given by:

$$\begin{bmatrix} x + \frac{-\dot{\psi} v \sin(\psi) + \left(\Delta t \dot{\psi} + \psi\right) v \cos\left(\Delta t \dot{\psi} + \psi\right)}{\dot{\psi}^2} \\ y + \frac{\dot{\psi} v \cos(\psi) - \left(\Delta t \dot{\psi} + \psi\right) v \sin\left(\Delta t \dot{\psi} + \psi\right)}{\dot{\psi}^2} \\ \Delta t \dot{\psi} + \psi \\ v \\ \dot{\psi} \end{bmatrix} \tag{F.14}$$

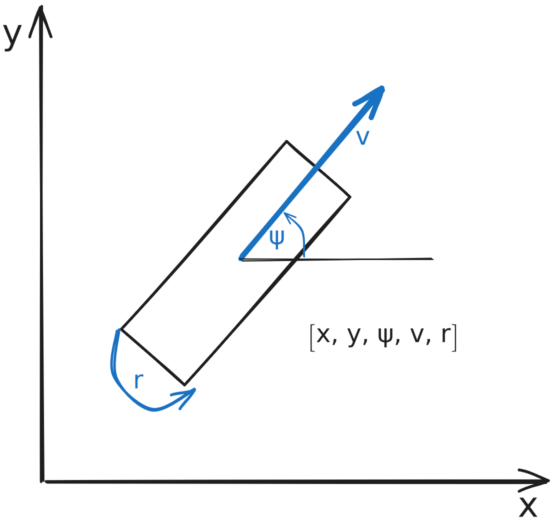


**Figure F.3:** Constant Turn Rate and Velocity Motion Model

### F.3.2. Noise Model

Derived using a similar method to section F.4.

$$Q_k^{CTRV} = \int_0^T e^{f'_{CTRV}\tau} \tilde{Q}_{CTRV} e^{f'_{CTRV}{}^T \tau} d\tau \tag{F.15}$$

$$\begin{bmatrix} \Delta t^3 \left( \frac{\Delta t^2 \sigma_{\dot{\omega}}^2 v^2 \sin^2(\psi)}{20} + \frac{\sigma_{\dot{v}}^2 \cos^2(\psi)}{3} \right) & \frac{\Delta t^3 (-3\Delta t^2 \sigma_{\dot{\omega}}^2 v^2 + 20\sigma_{\dot{v}}^2) \sin(2\psi)}{120} & -\frac{\Delta t^4 \sigma_{\dot{\omega}}^2 v \sin(\psi)}{8} & \frac{\Delta t^2 \sigma_{\dot{v}}^2 \cos(\psi)}{2} & -\frac{\Delta t^3 \sigma_{\dot{\omega}}^2 v \sin(\psi)}{6} \\ \frac{\Delta t^3 (-3\Delta t^2 \sigma_{\dot{\omega}}^2 v^2 + 20\sigma_{\dot{v}}^2) \sin(2\psi)}{120} & \Delta t^3 \left( \frac{\Delta t^2 \sigma_{\dot{\omega}}^2 v^2 \cos^2(\psi)}{20} + \frac{\sigma_{\dot{v}}^2 \sin^2(\psi)}{3} \right) & \frac{\Delta t^4 \sigma_{\dot{\omega}}^2 v \cos(\psi)}{8} & \frac{\Delta t^2 \sigma_{\dot{v}}^2 \sin(\psi)}{2} & \frac{\Delta t^3 \sigma_{\dot{\omega}}^2 v \cos(\psi)}{6} \\ -\frac{\Delta t^4 \sigma_{\dot{\omega}}^2 v \sin(\psi)}{8} & \frac{\Delta t^4 \sigma_{\dot{\omega}}^2 v \cos(\psi)}{8} & \frac{\Delta t^3 \sigma_{\dot{\omega}}^2}{3} & 0 & \frac{\Delta t^2 \sigma_{\dot{\omega}}^2}{2} \\ \frac{\Delta t^2 \sigma_{\dot{v}}^2 \cos(\psi)}{2} & \frac{\Delta t^2 \sigma_{\dot{v}}^2 \sin(\psi)}{2} & 0 & \Delta t \sigma_{\dot{v}}^2 & 0 \\ -\frac{\Delta t^3 \sigma_{\dot{\omega}}^2 v \sin(\psi)}{6} & \frac{\Delta t^3 \sigma_{\dot{\omega}}^2 v \cos(\psi)}{6} & \frac{\Delta t^2 \sigma_{\dot{\omega}}^2}{2} & 0 & \Delta t \sigma_{\dot{\omega}}^2 \end{bmatrix} \tag{F.16}$$

## F.4. Constant Turn Rate and Acceleration

The derivation of this model is based on [11].

the state vector for the constant turn rate and acceleration motion model is given by:

$$\mathbf{x_k} = \begin{bmatrix} x \\ y \\ \psi \\ v \\ \dot{\psi} \\ a \end{bmatrix} \tag{F.17}$$

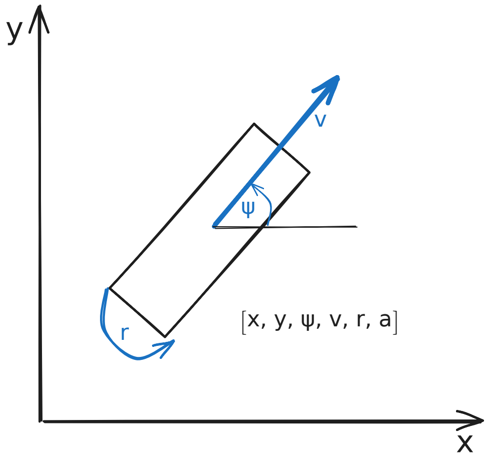


**Figure F.4:** Constant Turn Rate and Acceleration Motion Model

### F.4.1. Transition Model

the continuous-time transition model for the constant turn rate and acceleration motion model is given by:

$$f_{\text{CTRA}} = \begin{bmatrix} v\cos(\psi) \\ v\sin(\psi) \\ \dot{\psi} \\ a \\ 0 \\ 0 \end{bmatrix} \tag{F.18}$$

The discrete-time transition functions are given by:

$$\begin{bmatrix} x + \frac{-\dot{\psi}v\sin(\psi) - a\cos(\psi) + a\cos\left(\Delta t\dot{\psi} + \psi\right) + \left(\Delta t\dot{\psi}a + \dot{\psi}v\right)\sin\left(\Delta t\dot{\psi} + \psi\right)}{\dot{\psi}^2} \\ y + \frac{\dot{\psi}v\cos(\psi) - a\sin(\psi) + a\sin\left(\Delta t\dot{\psi} + \psi\right) + \left(-\Delta t\dot{\psi}a - \dot{\psi}v\right)\cos\left(\Delta t\dot{\psi} + \psi\right)}{\dot{\psi}^2} \\ \Delta t\dot{\psi} + \psi \\ \Delta t a + v \\ \dot{\psi} \\ a \end{bmatrix} \tag{F.19}$$

### F.4.2. Noise Model

The derivation of the discrete-time process noise covariance matrix, $Q_k^{CTRA}$, begins with the general formula for converting a continuous-time noise process to a discrete-time one:

$$Q_k^{CTRA} = \int_0^T e^{f'_{CTRA}\tau}\tilde{Q}_{CTRA}e^{{f'_{CTRA}}^T\tau}d\tau \tag{F.20}$$

where $f'_{CTRA}$ is the gradient of the motion model prediction function, and $\tilde{Q}_{CTRA} = \Gamma Q_{CTRA}\Gamma^T$. The continuous-time noise spectral density matrix $Q_{CTRA}$ and the mapping matrix $\Gamma$ are given as:

$$Q_{CTRA} = \begin{pmatrix} \sigma_{\ddot{\psi}}^2 & 0 \\ 0 & \sigma_{\dot{a}}^2 \end{pmatrix}, \quad \Gamma = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

The gradient of the continuous-time prediction function, $f'_{CTRA} = Jacf_{CTRA}$, is:

$$f'_{CTRA} = \begin{bmatrix} 0 & 0 & -v\sin(\psi) & \cos(\psi) & 0 & 0 \\ 0 & 0 & v\cos(\psi) & \sin(\psi) & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Since $f'_{CTRA}$ is a nilpotent matrix, its exponential can be computed from a finite series expansion:

$$e^{f'_{CTRA}\tau} = I + \tau f'_{CTRA} + \frac{\tau^2}{2}(f'_{CTRA})^2$$

The square of the gradient matrix is:

The product inside the integral of Eq. F.20 is then computed:

$$e^{f'_{CTRA}\tau}\tilde{Q}_{CTRA}e^{{f'_{CTRA}}^{T}\tau} =$$

$$\begin{bmatrix} \frac{\tau^4\left(\sigma_{\dot{\omega}}^2 v^2\sin^2(\psi)+\sigma_{\dot{a}}^2\cos^2(\psi)\right)}{4} & \frac{\tau^4\left(-\sigma_{\dot{\omega}}^2 v^2+\sigma_{\dot{a}}^2\right)\sin(2\psi)}{8} & -\frac{\sigma_{\dot{\omega}}^2\tau^3 v\sin(\psi)}{2} & \frac{\sigma_{\dot{a}}^2\tau^3\cos(\psi)}{2} & -\frac{\sigma_{\dot{\omega}}^2\tau^2 v\sin(\psi)}{2} & \frac{\sigma_{\dot{a}}^2\tau^2\cos(\psi)}{2} \\ \frac{\tau^4\left(-\sigma_{\dot{\omega}}^2 v^2+\sigma_{\dot{a}}^2\right)\sin(2\psi)}{8} & \frac{\tau^4\left(\sigma_{\dot{\omega}}^2 v^2\cos^2(\psi)+\sigma_{\dot{a}}^2\sin^2(\psi)\right)}{4} & \frac{\sigma_{\dot{\omega}}^2\tau^3 v\cos(\psi)}{2} & \frac{\sigma_{\dot{a}}^2\tau^3\sin(\psi)}{2} & \frac{\sigma_{\dot{\omega}}^2\tau^2 v\cos(\psi)}{2} & \frac{\sigma_{\dot{a}}^2\tau^2\sin(\psi)}{2} \\ -\frac{\sigma_{\dot{\omega}}^2\tau^3 v\sin(\psi)}{2} & \frac{\sigma_{\dot{\omega}}^2\tau^3 v\cos(\psi)}{2} & \sigma_{\dot{\omega}}^2\tau^2 & 0 & \sigma_{\dot{\omega}}^2\tau & 0 \\ \frac{\sigma_{\dot{a}}^2\tau^3\cos(\psi)}{2} & \frac{\sigma_{\dot{a}}^2\tau^3\sin(\psi)}{2} & 0 & \sigma_{\dot{a}}^2\tau^2 & 0 & \sigma_{\dot{a}}^2\tau \\ -\frac{\sigma_{\dot{\omega}}^2\tau^2 v\sin(\psi)}{2} & \frac{\sigma_{\dot{\omega}}^2\tau^2 v\cos(\psi)}{2} & \sigma_{\dot{\omega}}^2\tau & 0 & \sigma_{\dot{\omega}}^2 & 0 \\ \frac{\sigma_{\dot{a}}^2\tau^2\cos(\psi)}{2} & \frac{\sigma_{\dot{a}}^2\tau^2\sin(\psi)}{2} & 0 & \sigma_{\dot{a}}^2\tau & 0 & \sigma_{\dot{a}}^2 \end{bmatrix}$$

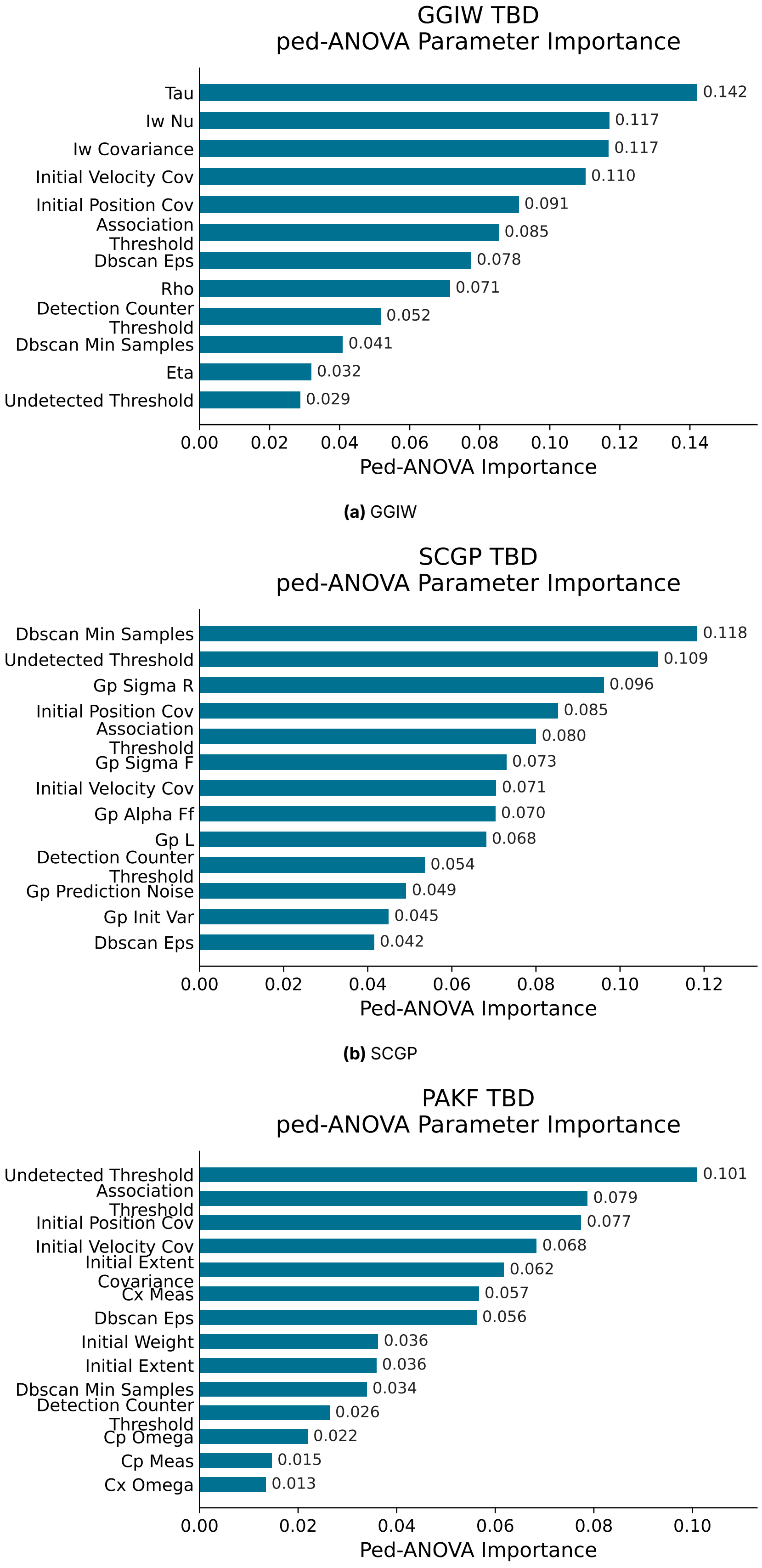
Integrating this matrix from 0 to $\Delta t$ yields the final discrete-time process noise covariance matrix $Q_k^{CTRA}$:

$$\begin{bmatrix} \frac{\Delta t^5\left(\sigma_{\dot{\omega}}^2 v^2\sin^2(\psi)+\sigma_{\dot{a}}^2\cos^2(\psi)\right)}{20} & \frac{\Delta t^5\left(-\sigma_{\dot{\omega}}^2 v^2+\sigma_{\dot{a}}^2\right)\sin(2\psi)}{40} & -\frac{\Delta t^4\sigma_{\dot{\omega}}^2 v\sin(\psi)}{8} & \frac{\Delta t^4\sigma_{\dot{a}}^2\cos(\psi)}{8} & -\frac{\Delta t^3\sigma_{\dot{\omega}}^2 v\sin(\psi)}{6} & \frac{\Delta t^3\sigma_{\dot{a}}^2\cos(\psi)}{6} \\ \frac{\Delta t^5\left(-\sigma_{\dot{\omega}}^2 v^2+\sigma_{\dot{a}}^2\right)\sin(2\psi)}{40} & \frac{\Delta t^5\left(\sigma_{\dot{\omega}}^2 v^2\cos^2(\psi)+\sigma_{\dot{a}}^2\sin^2(\psi)\right)}{20} & \frac{\Delta t^4\sigma_{\dot{\omega}}^2 v\cos(\psi)}{8} & \frac{\Delta t^4\sigma_{\dot{a}}^2\sin(\psi)}{8} & \frac{\Delta t^3\sigma_{\dot{\omega}}^2 v\cos(\psi)}{6} & \frac{\Delta t^3\sigma_{\dot{a}}^2\sin(\psi)}{6} \\ -\frac{\Delta t^4\sigma_{\dot{\omega}}^2 v\sin(\psi)}{8} & \frac{\Delta t^4\sigma_{\dot{\omega}}^2 v\cos(\psi)}{8} & \frac{\Delta t^3\sigma_{\dot{\omega}}^2}{3} & 0 & \frac{\Delta t^2\sigma_{\dot{\omega}}^2}{2} & 0 \\ \frac{\Delta t^4\sigma_{\dot{a}}^2\cos(\psi)}{8} & \frac{\Delta t^4\sigma_{\dot{a}}^2\sin(\psi)}{8} & 0 & \frac{\Delta t^3\sigma_{\dot{a}}^2}{3} & 0 & \frac{\Delta t^2\sigma_{\dot{a}}^2}{2} \\ -\frac{\Delta t^3\sigma_{\dot{\omega}}^2 v\sin(\psi)}{6} & \frac{\Delta t^3\sigma_{\dot{\omega}}^2 v\cos(\psi)}{6} & \frac{\Delta t^2\sigma_{\dot{\omega}}^2}{2} & 0 & \Delta t\sigma_{\dot{\omega}}^2 & 0 \\ \frac{\Delta t^3\sigma_{\dot{a}}^2\cos(\psi)}{6} & \frac{\Delta t^3\sigma_{\dot{a}}^2\sin(\psi)}{6} & 0 & \frac{\Delta t^2\sigma_{\dot{a}}^2}{2} & 0 & \Delta t\sigma_{\dot{a}}^2 \end{bmatrix}$$

# G

# Parameter Study Results

## G.1. Track-by-Detection



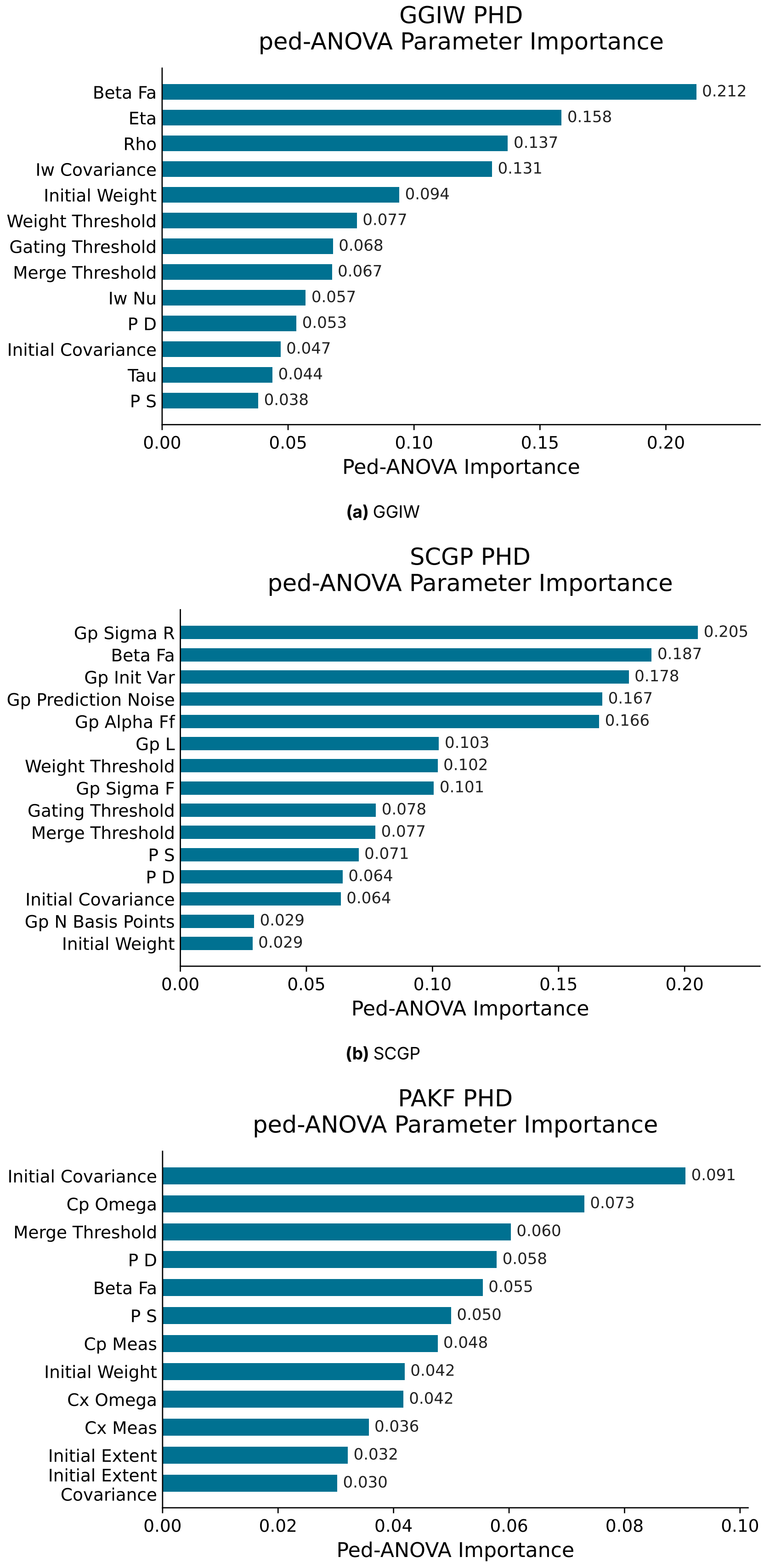
**(a)** GGIW

**(b)** SCGP

**(c)** PAKF

**Figure G.1:** ANOVA feature importance for the TBD parameter study across three methods. Each panel's height is constrained so the full figure fits within a page.

### G.1.1. Probablity Hypothesis Density



(a) GGIW

(b) SCGP

(c) PAKF

**Figure G.2:** ANOVA feature importance for the PHD parameter study (SCGP, GGIW, PAKF). Each panel's height is constrained so the full figure fits within a page.

## G.2. Poisson Multi-Bernoulli Mixture

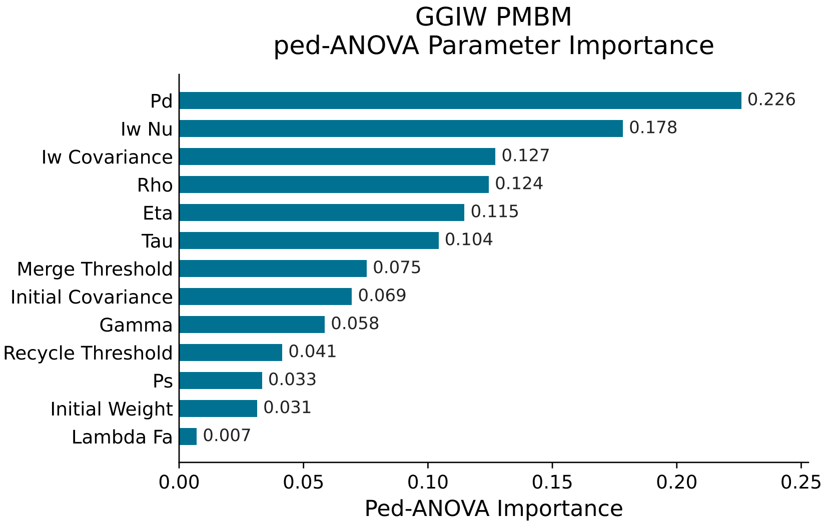


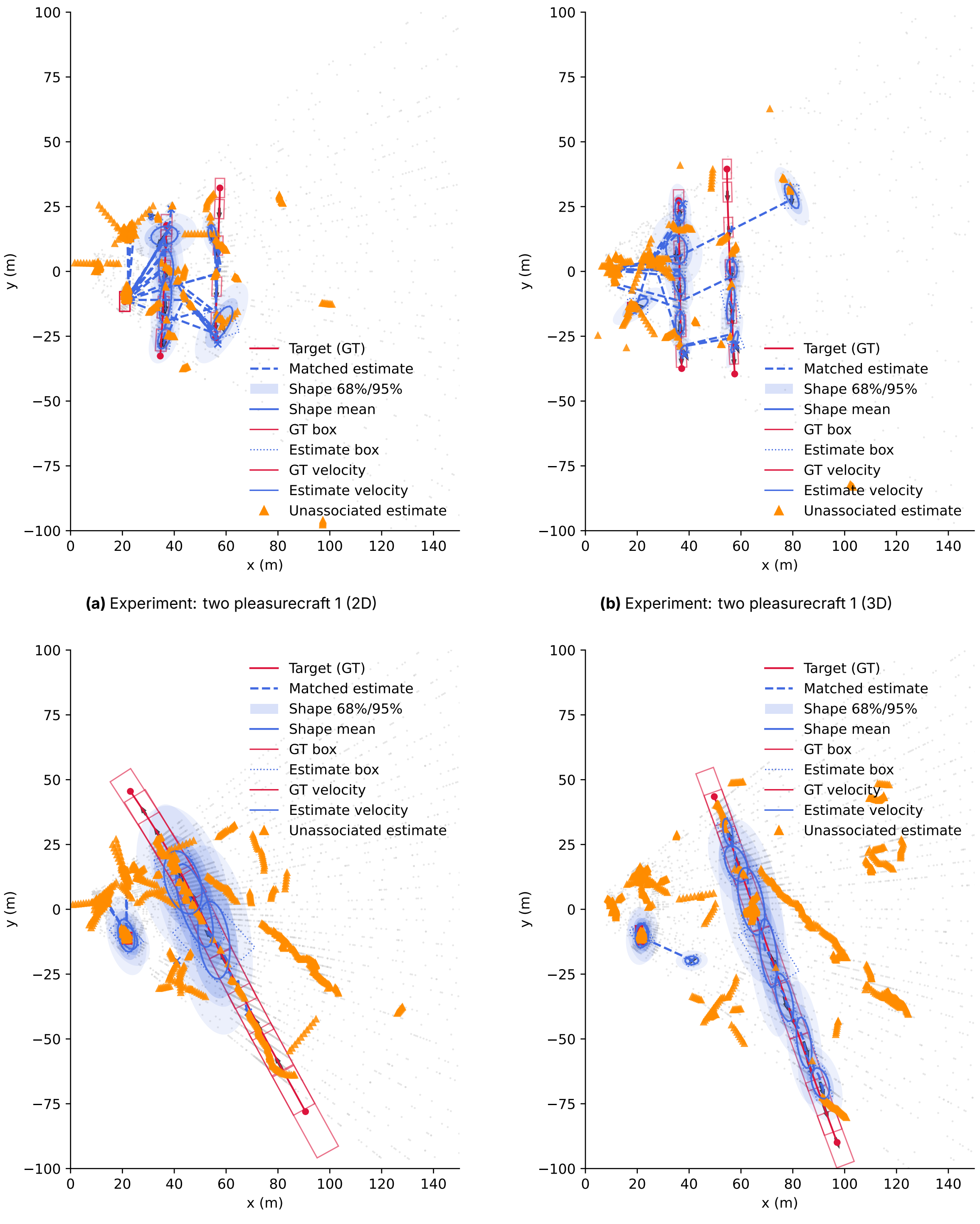
**Figure G.3:** ANOVA feature importance for the GGIW PMBM parameter study.

# H

# Resulting Estimate Plots

## H.1. Track by Detection

### H.1.1. Gaussian inverse Wishart



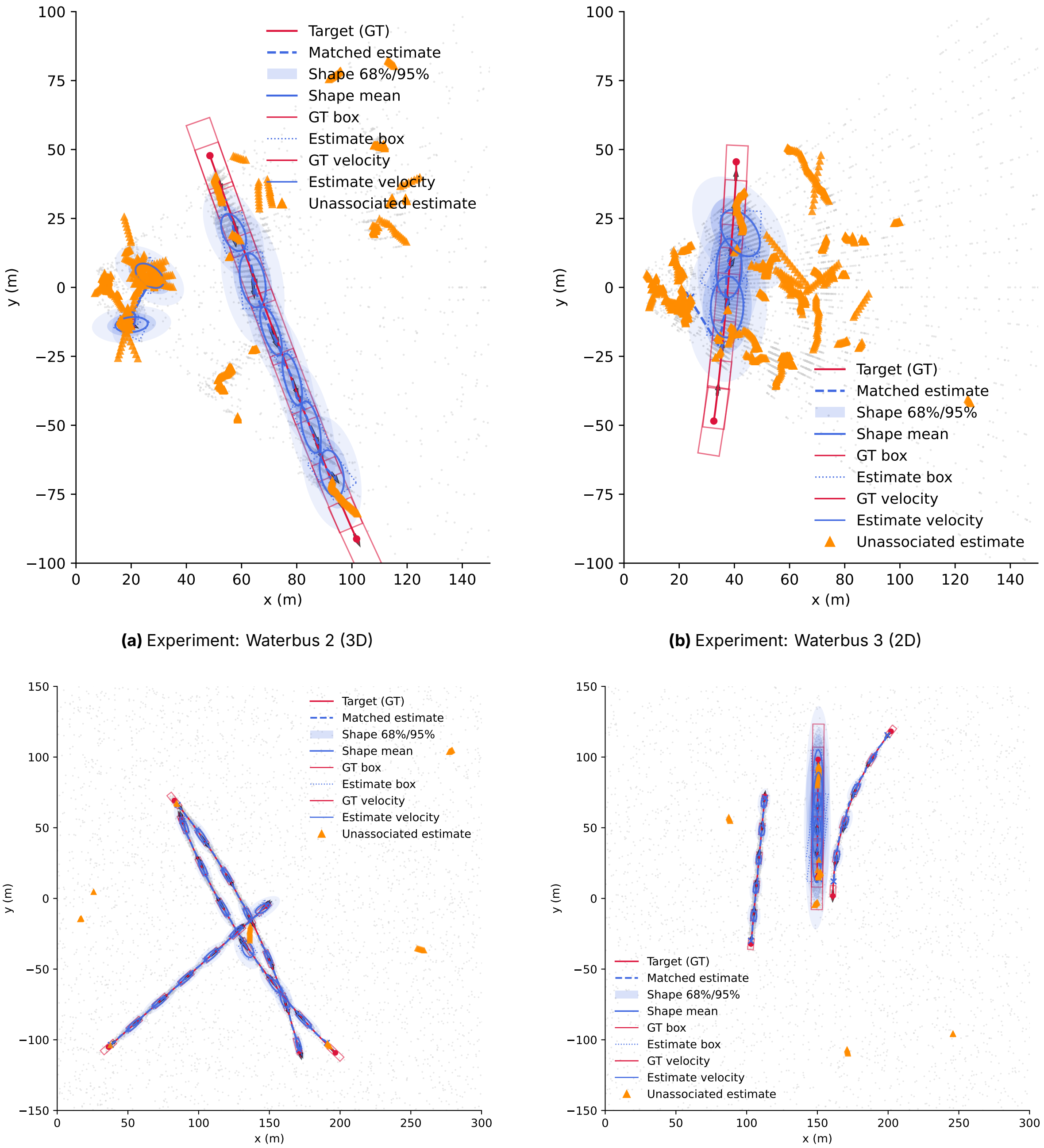
**(a)** Experiment: two pleasurecraft 1 (2D)

**(b)** Experiment: two pleasurecraft 1 (3D)

**(c)** Experiment: Waterbus 1 (2D)

**(d)** Experiment: Waterbus 2 (2D)

**Figure H.1:** GIW estimated tracks — track-by-detection results (1/3). Each subfigure shows estimates for the named experiment or simulation.



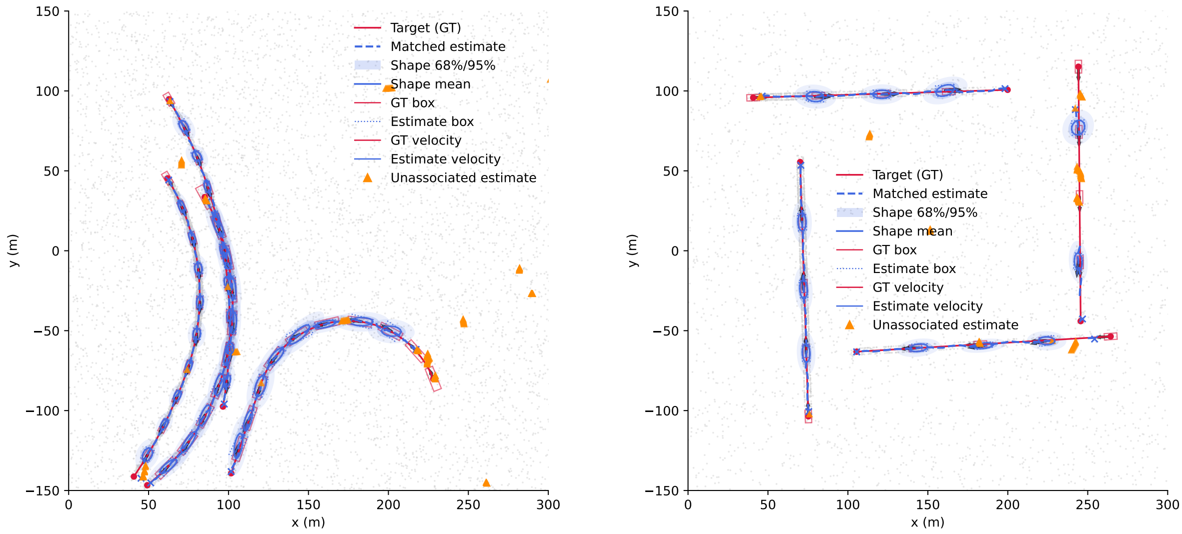
**(a)** Experiment: Waterbus 2 (3D)

**(b)** Experiment: Waterbus 3 (2D)

**(c)** Simulated: 3 targets, complex motions

**(d)** Simulated: 3 targets, varying sizes

**Figure H.2:** GIW estimated tracks — track-by-detection results (2/3). Each subfigure shows estimates for the named experiment or simulation.

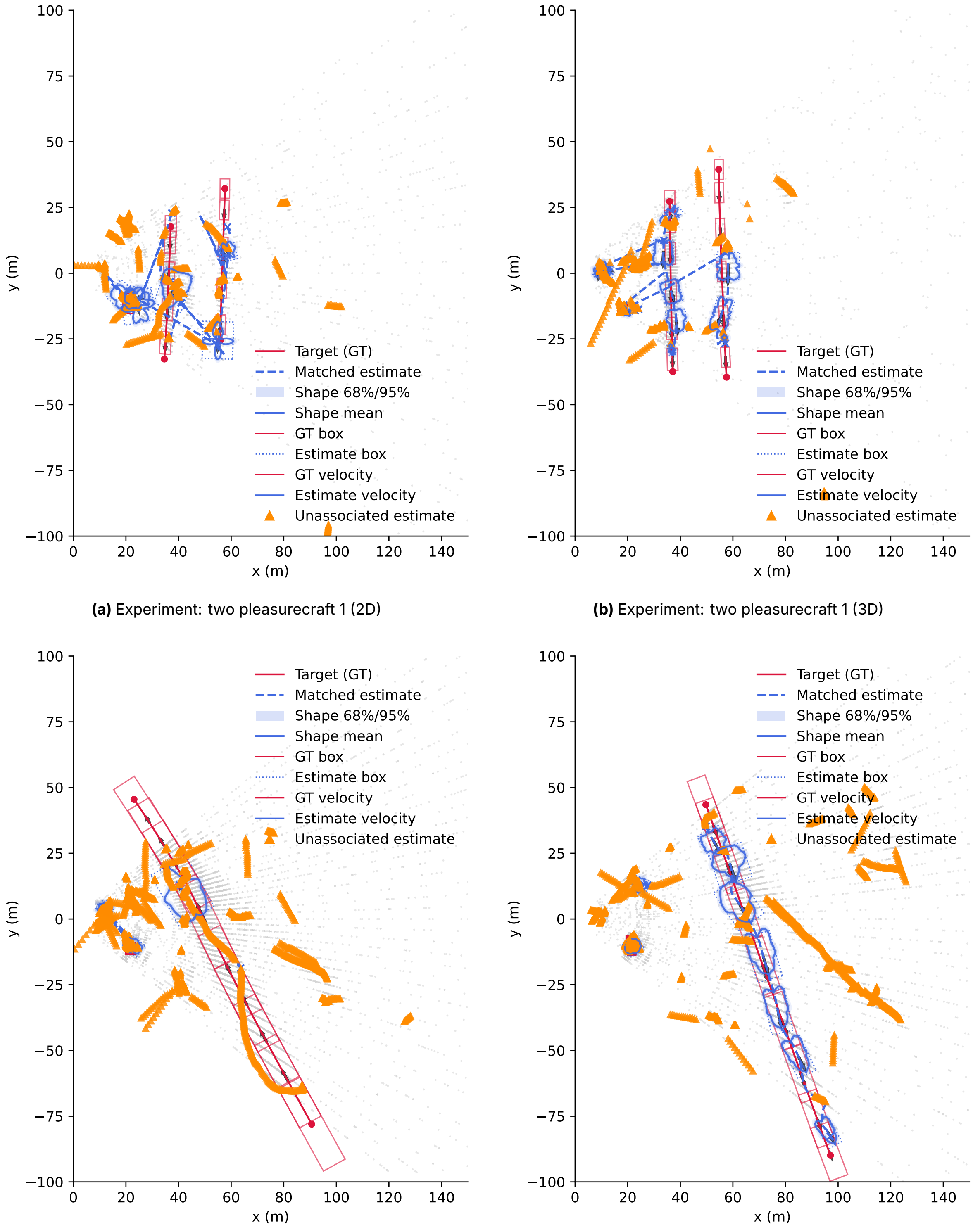


**(a)** Simulated: 4 targets, complex motions

**(b)** Simulated: 4 targets, rectangle CV

**Figure H.3:** GIW estimated tracks — track-by-detection results (3/3). Each subfigure shows estimates for the named experiment or simulation.

## H.1.2. Star Convex Gaussian Process



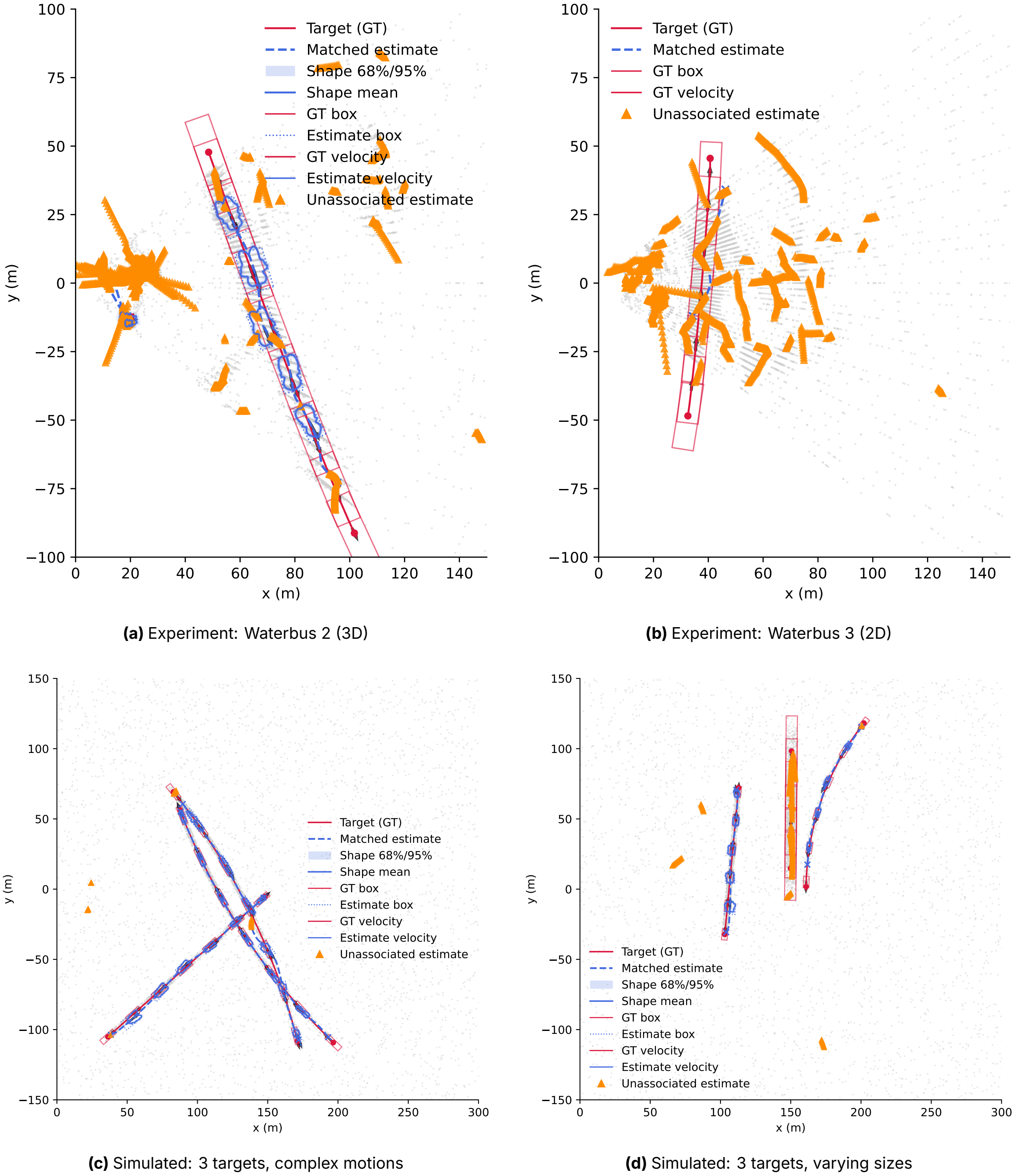
**(a)** Experiment: two pleasurecraft 1 (2D)

**(b)** Experiment: two pleasurecraft 1 (3D)

**(c)** Experiment: Waterbus 1 (2D)

**(d)** Experiment: Waterbus 2 (2D)

**Figure H.4:** SCGP estimated tracks — track-by-detection results (1/3). Each subfigure shows estimates for the named experiment or simulation.



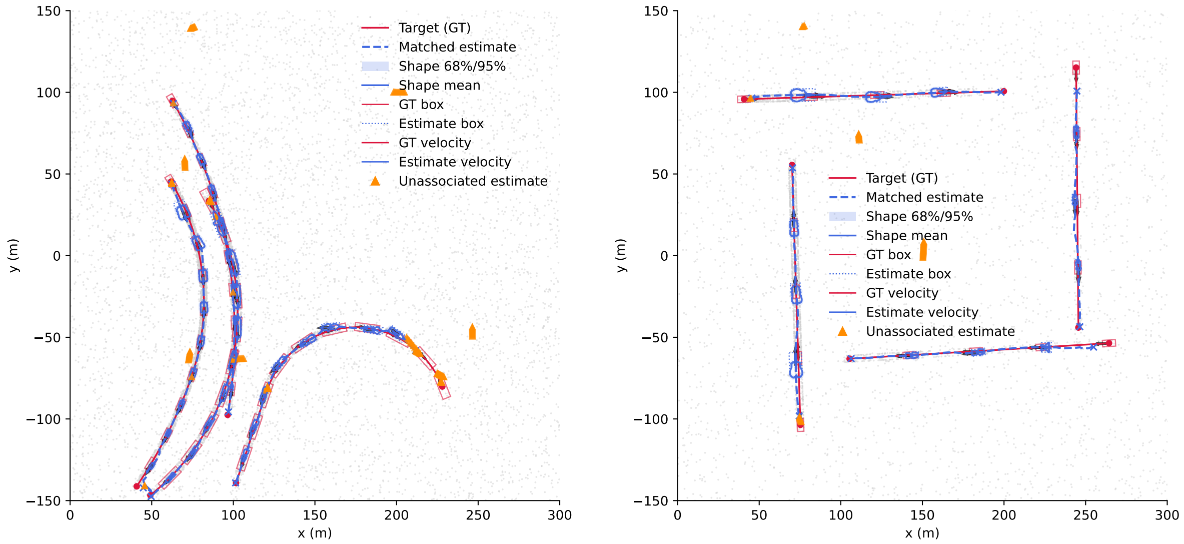
**(a)** Experiment: Waterbus 2 (3D)

**(b)** Experiment: Waterbus 3 (2D)

**(c)** Simulated: 3 targets, complex motions

**(d)** Simulated: 3 targets, varying sizes

**Figure H.5:** SCGP estimated tracks — track-by-detection results (2/3). Each subfigure shows estimates for the named experiment or simulation.

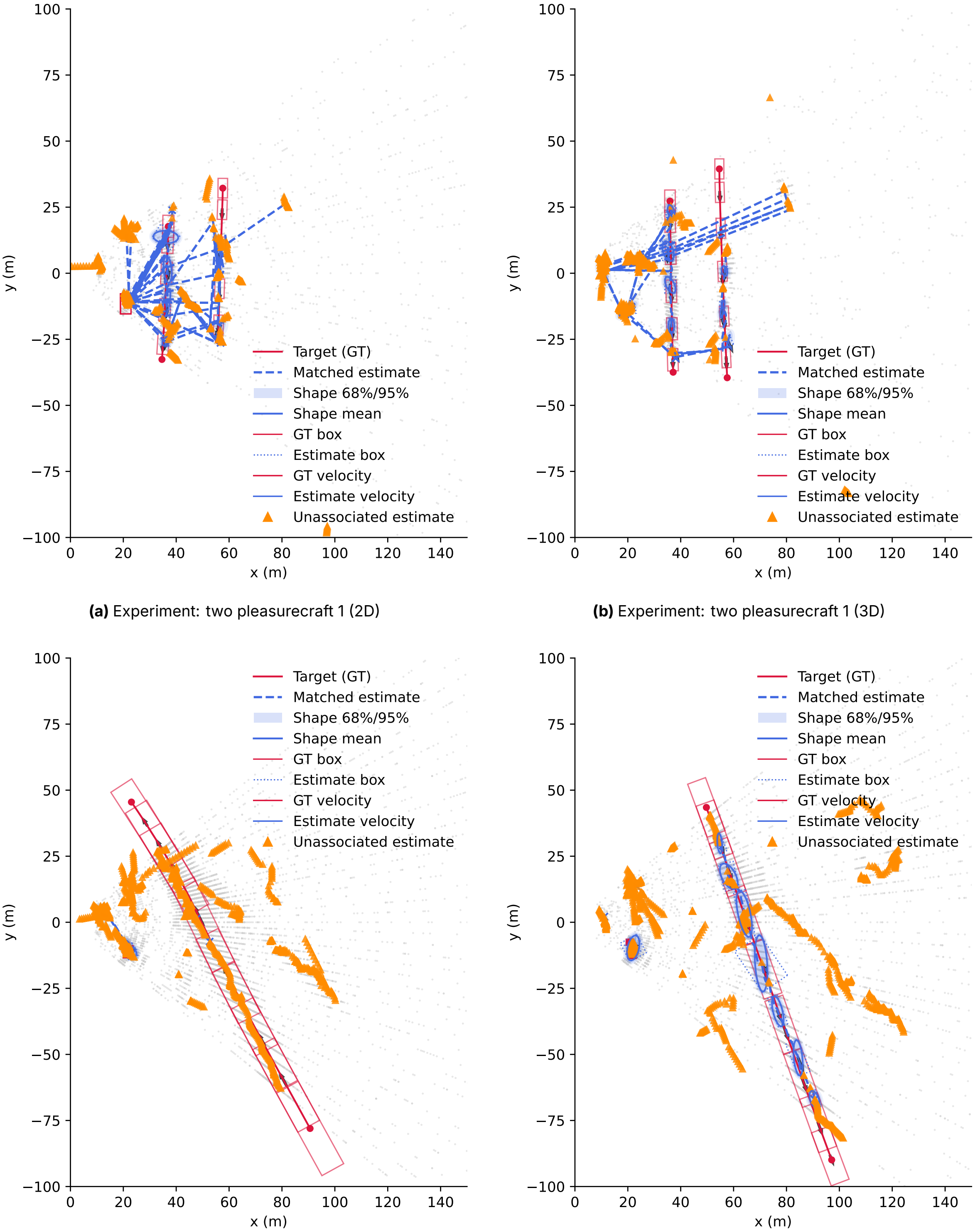


**(a)** Simulated: 4 targets, complex motions

**(b)** Simulated: 4 targets, rectangle CV

**Figure H.6:** SCGP estimated tracks — track-by-detection results (3/3). Each subfigure shows estimates for the named experiment or simulation.

### H.1.3. PAKF



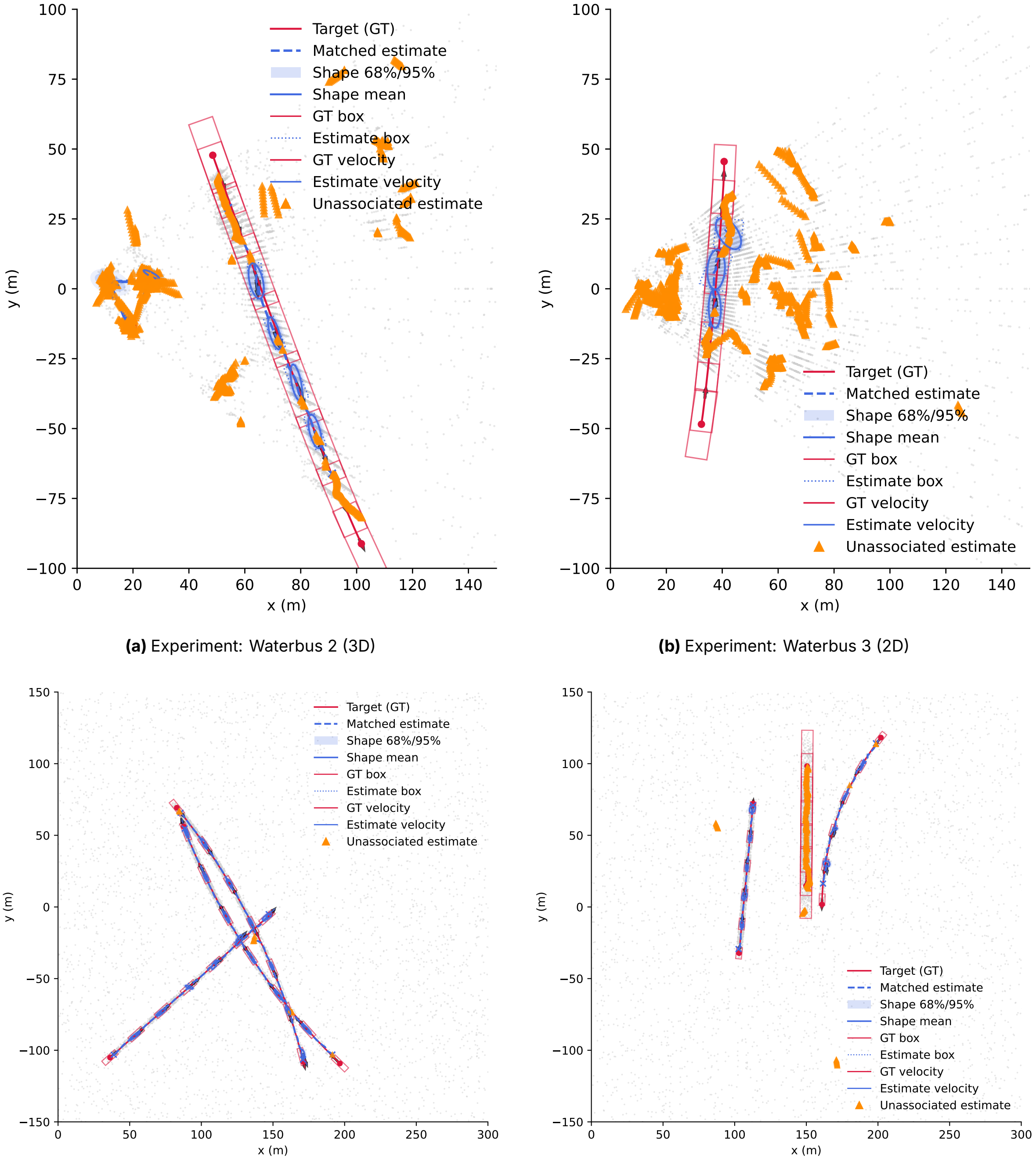
**(a)** Experiment: two pleasurecraft 1 (2D)

**(b)** Experiment: two pleasurecraft 1 (3D)

**(c)** Experiment: Waterbus 1 (2D)

**(d)** Experiment: Waterbus 2 (2D)

**Figure H.7:** PAKF estimated tracks — track-by-detection results (1/3). Each subfigure shows estimates for the named experiment or simulation.



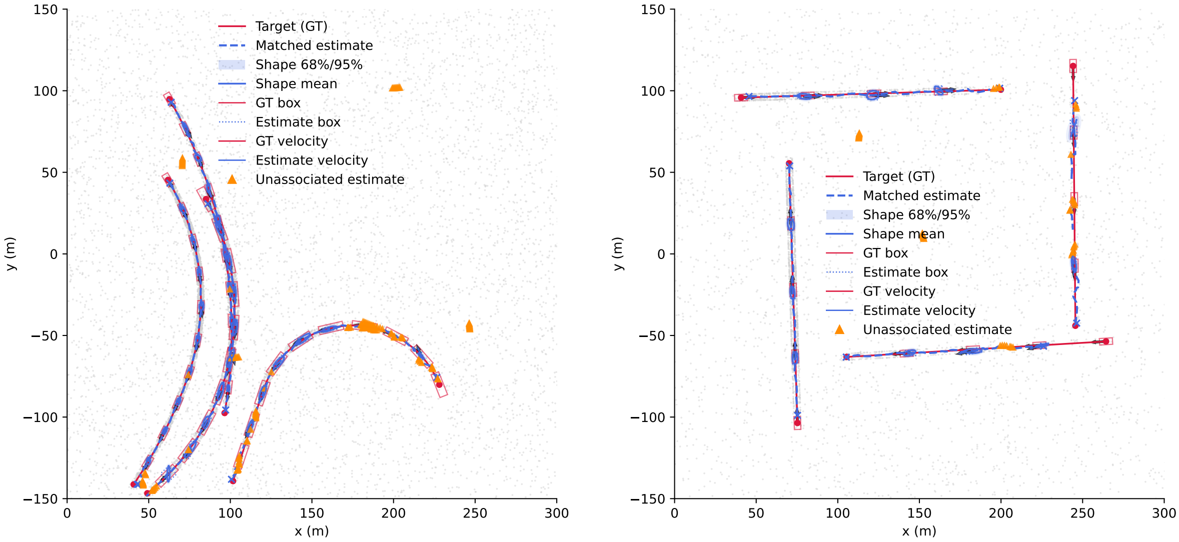
**(a)** Experiment: Waterbus 2 (3D)

**(b)** Experiment: Waterbus 3 (2D)

**(c)** Simulated: 3 targets, complex motions

**(d)** Simulated: 3 targets, varying sizes

**Figure H.8:** PAKF estimated tracks — track-by-detection results (2/3). Each subfigure shows estimates for the named experiment or simulation.



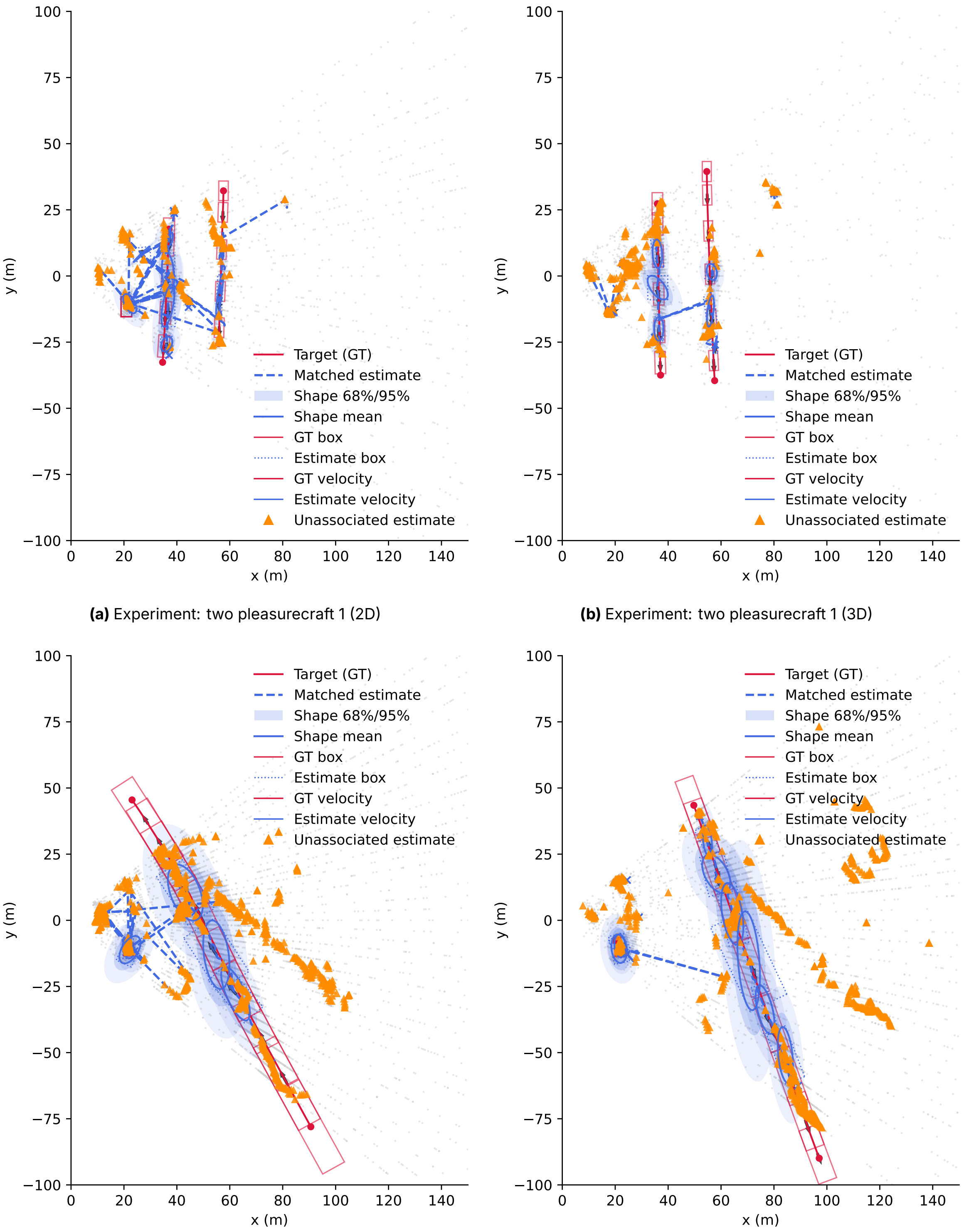
**(a)** Simulated: 4 targets, complex motions

**(b)** Simulated: 4 targets, rectangle CV

**Figure H.9:** PAKF estimated tracks — track-by-detection results (3/3). Each subfigure shows estimates for the named experiment or simulation.

## H.2. Probability Hypothesis Density Filter

### H.2.1. Gaussian Inverse Wishart



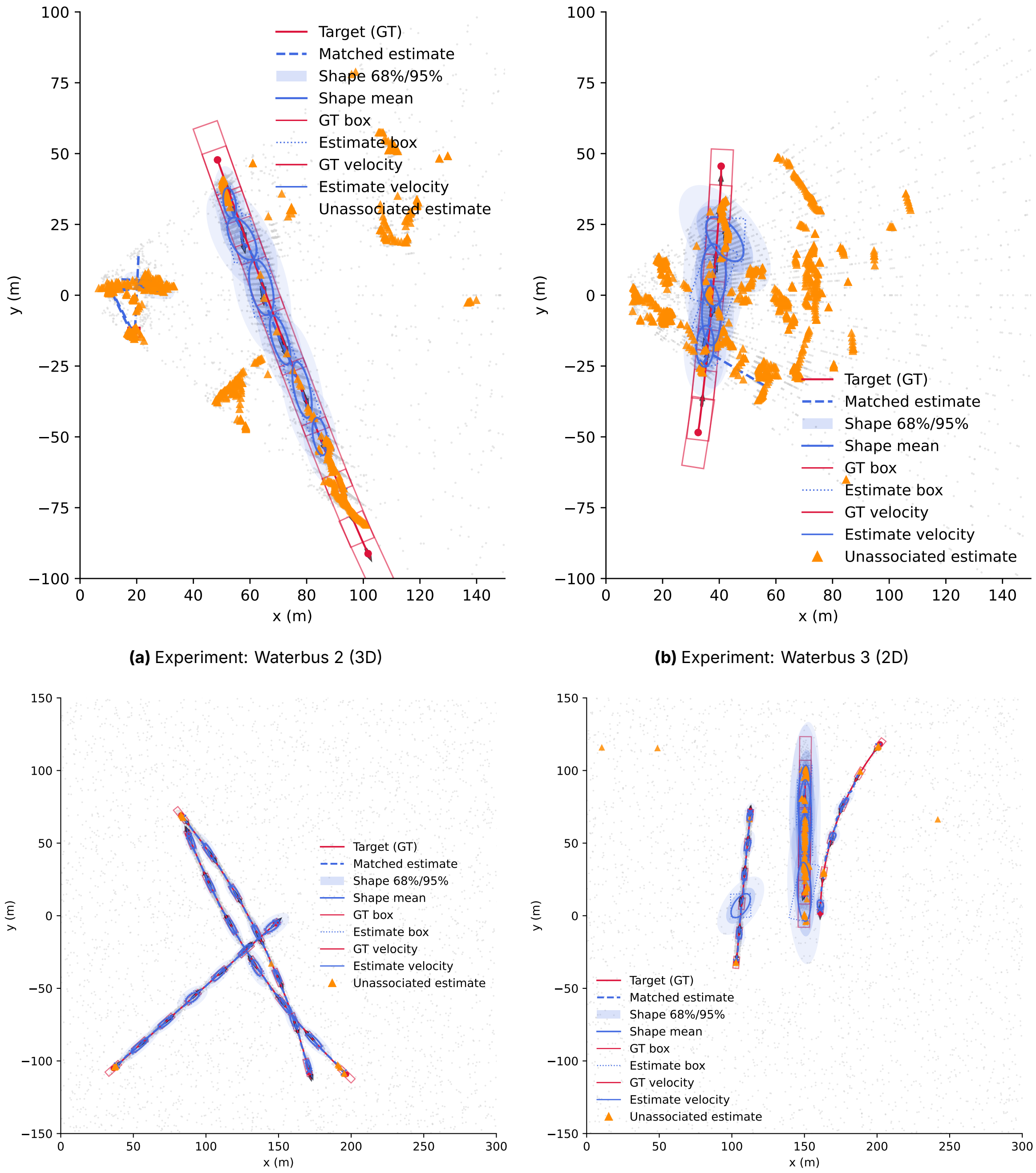
**(a)** Experiment: two pleasurecraft 1 (2D)

**(b)** Experiment: two pleasurecraft 1 (3D)

**(c)** Experiment: Waterbus 1 (2D)

**(d)** Experiment: Waterbus 2 (2D)

**Figure H.10:** PHD (GIW) estimated tracks — track-by-detection results (1/3). Each subfigure shows estimates for the named experiment or simulation.



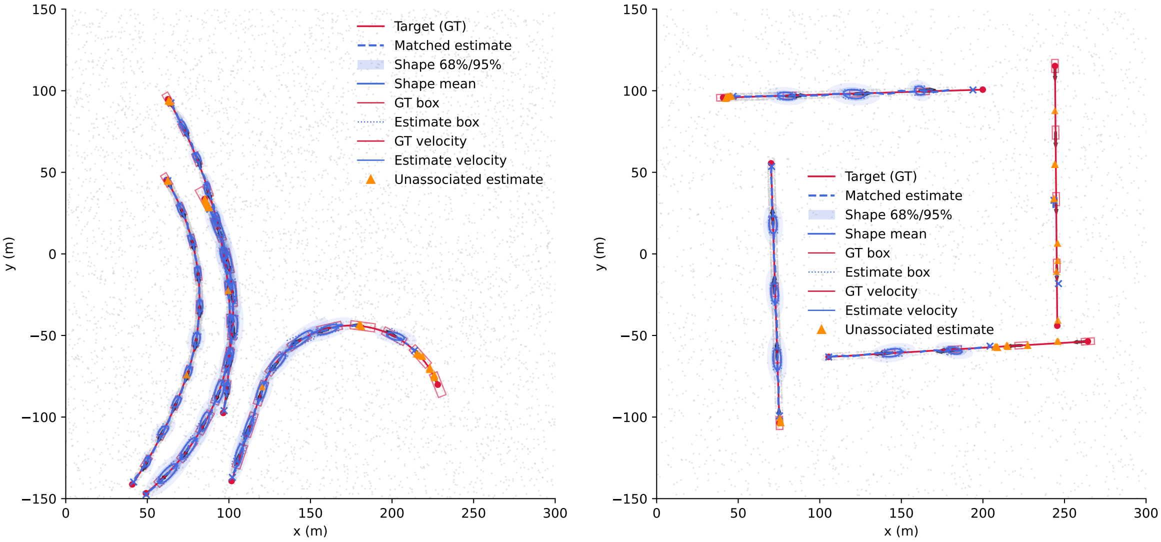
**(a)** Experiment: Waterbus 2 (3D)

**(b)** Experiment: Waterbus 3 (2D)

**(c)** Simulated: 3 targets, complex motions

**(d)** Simulated: 3 targets, varying sizes

**Figure H.11:** PHD (GIW) estimated tracks — track-by-detection results (2/3). Each subfigure shows estimates for the named experiment or simulation.

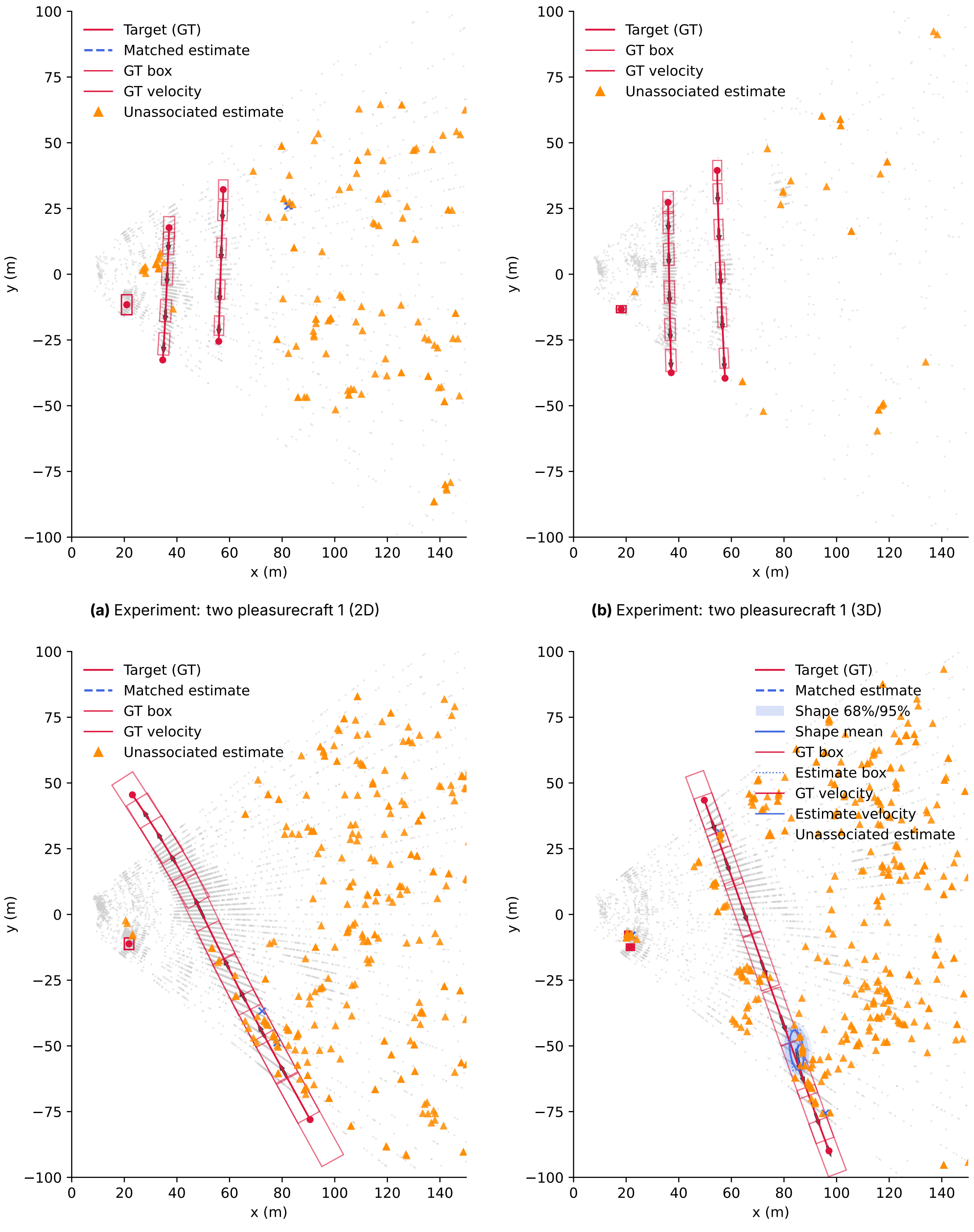


**(a)** Simulated: 4 targets, complex motions

**(b)** Simulated: 4 targets, rectangle CV

**Figure H.12:** PHD (GIW) estimated tracks — track-by-detection results (3/3). Remaining subfigures for the named simulations.

### H.2.2. Star Convex Gaussian Process



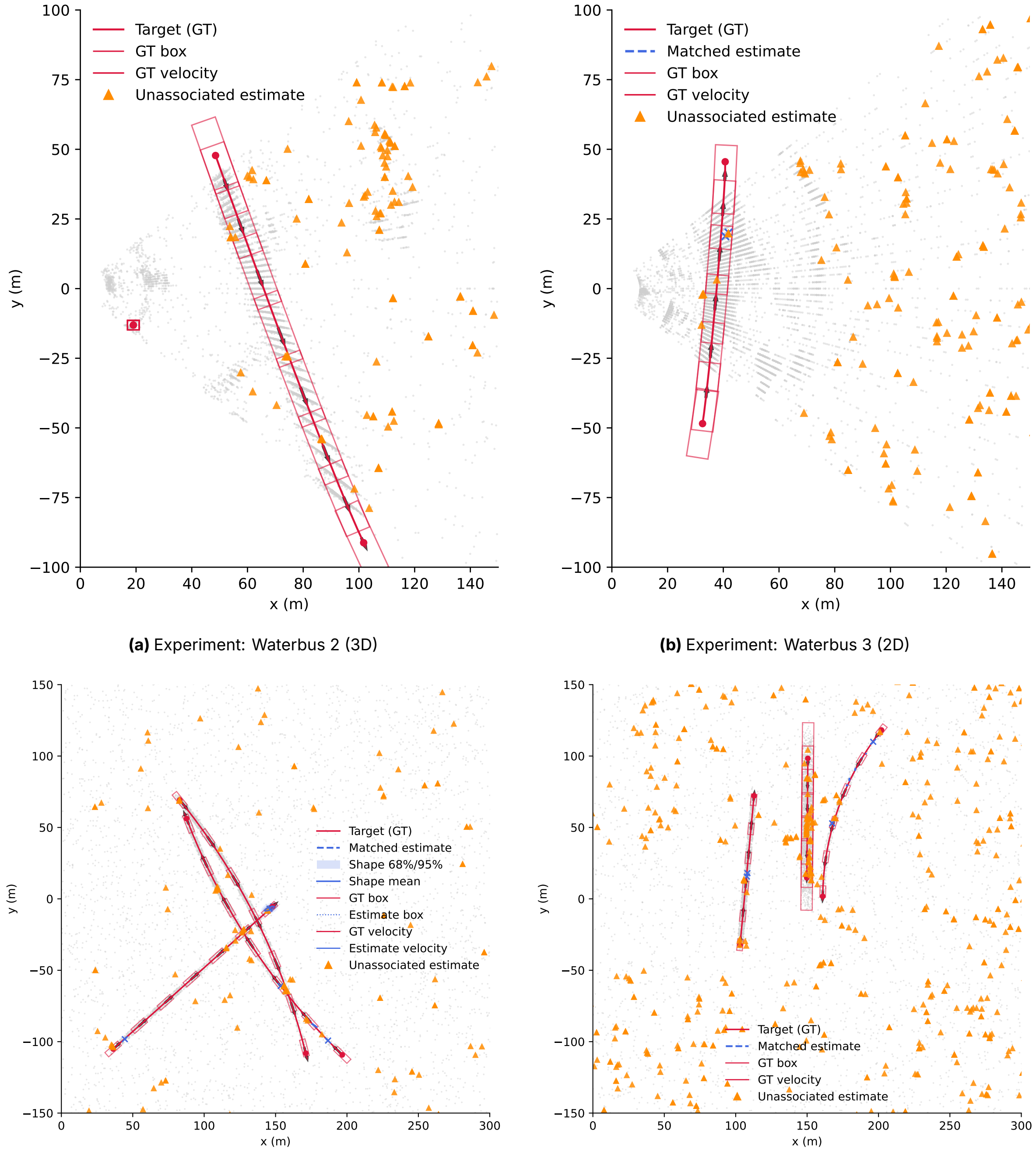
**(a)** Experiment: two pleasurecraft 1 (2D)

**(b)** Experiment: two pleasurecraft 1 (3D)

**(c)** Experiment: Waterbus 1 (2D)

**(d)** Experiment: Waterbus 2 (2D)

**Figure H.13:** PHD (SCGP) estimated tracks — track-by-detection results (1/3). Each subfigure shows estimates for the named experiment or simulation.



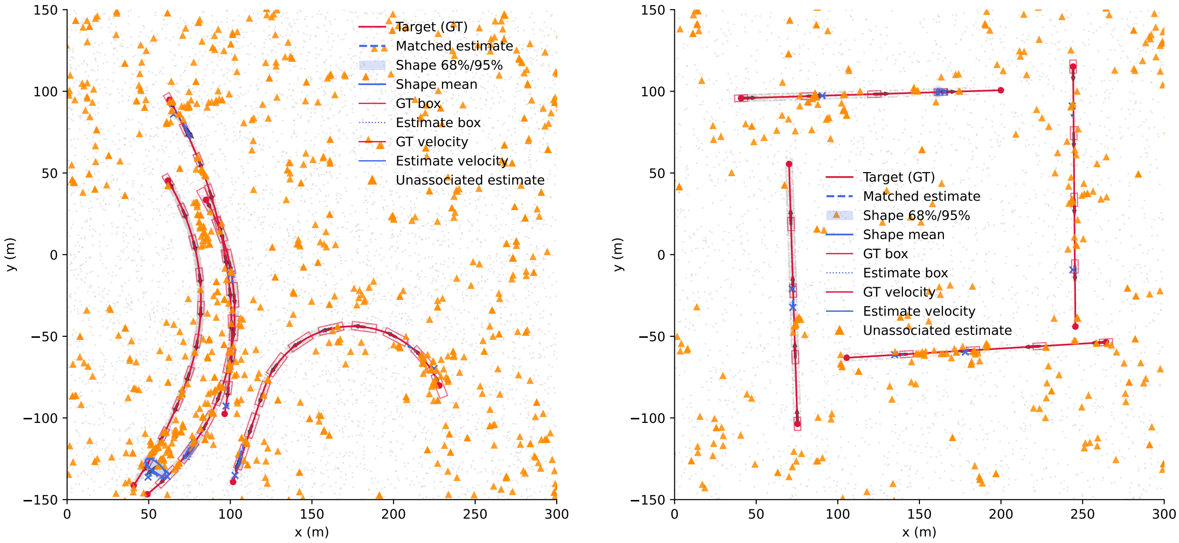
**(a)** Experiment: Waterbus 2 (3D)

**(b)** Experiment: Waterbus 3 (2D)

**(c)** Simulated: 3 targets, complex motions

**(d)** Simulated: 3 targets, varying sizes

**Figure H.14:** PHD (SCGP) estimated tracks — track-by-detection results (2/3). Each subfigure shows estimates for the named experiment or simulation.

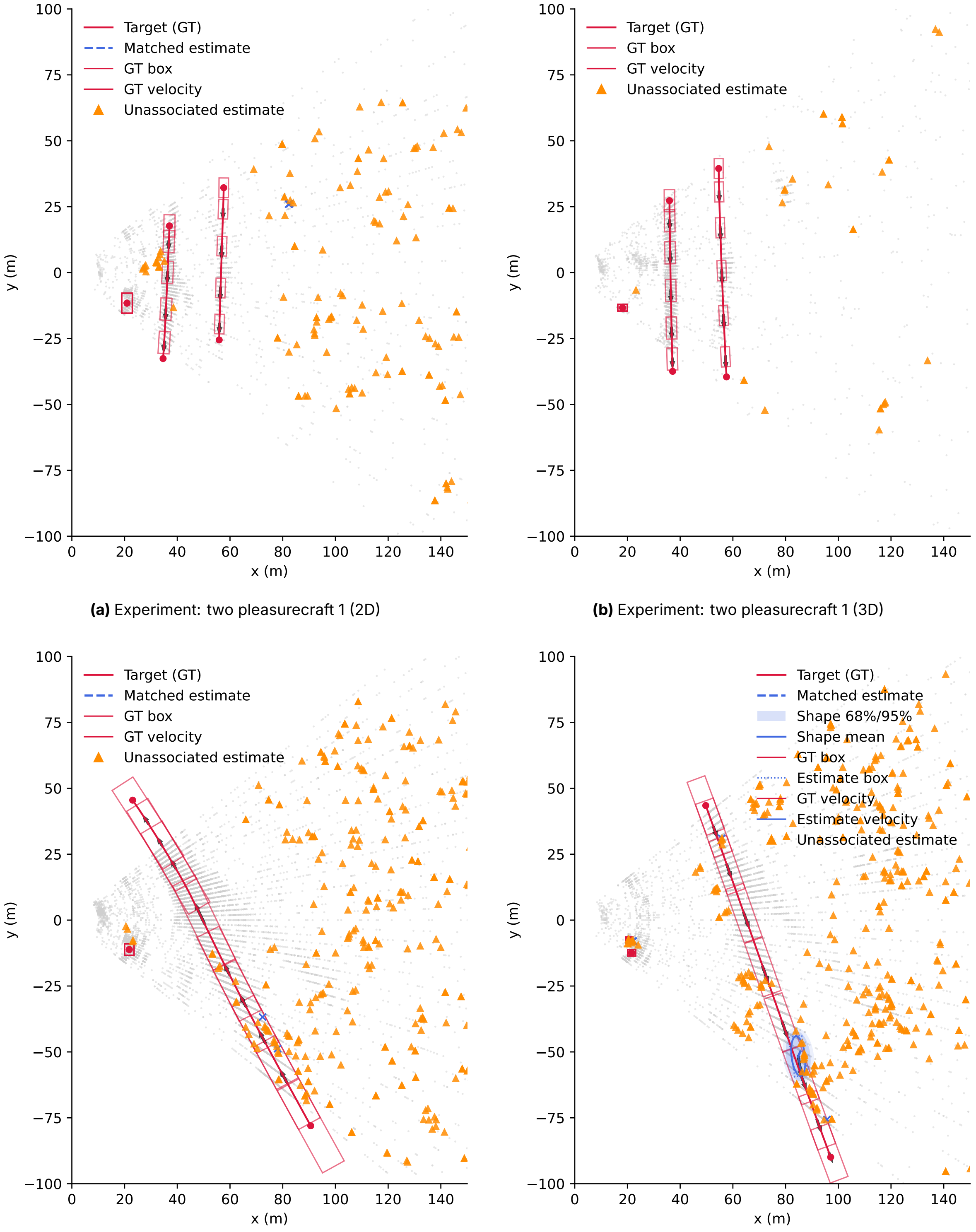


**(a)** Simulated: 4 targets, complex motions

**(b)** Simulated: 4 targets, rectangle CV

**Figure H.15:** PHD (SCGP) estimated tracks — track-by-detection results (3/3). Remaining subfigures for the named simulations.

### H.2.3. Principle Axis Kalman Filter



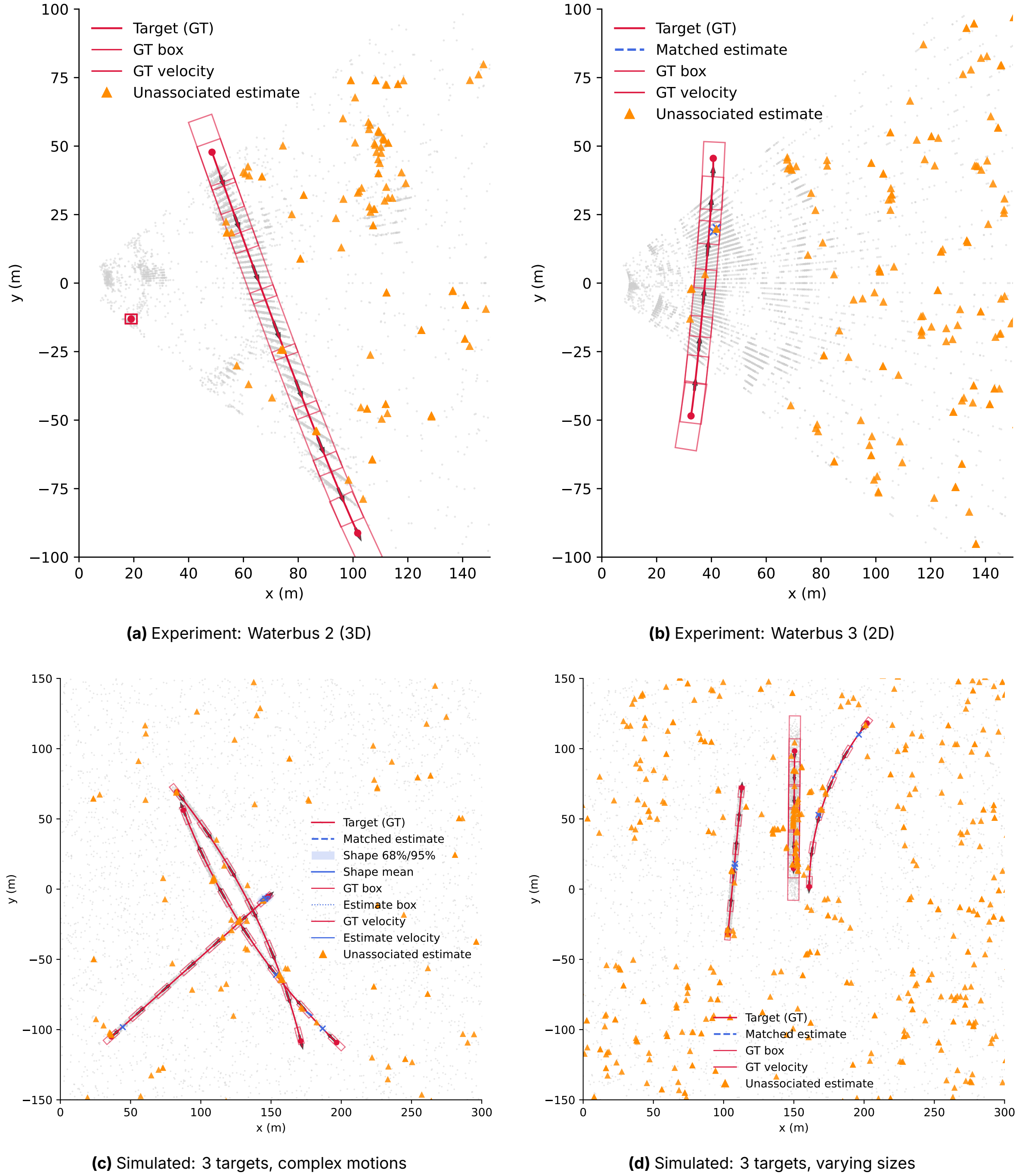
**(a)** Experiment: two pleasurecraft 1 (2D)

**(b)** Experiment: two pleasurecraft 1 (3D)

**(c)** Experiment: Waterbus 1 (2D)

**(d)** Experiment: Waterbus 2 (2D)

**Figure H.16:** PHD (PAKF) estimated tracks — track-by-detection results (1/3). Each subfigure shows estimates for the named experiment or simulation.



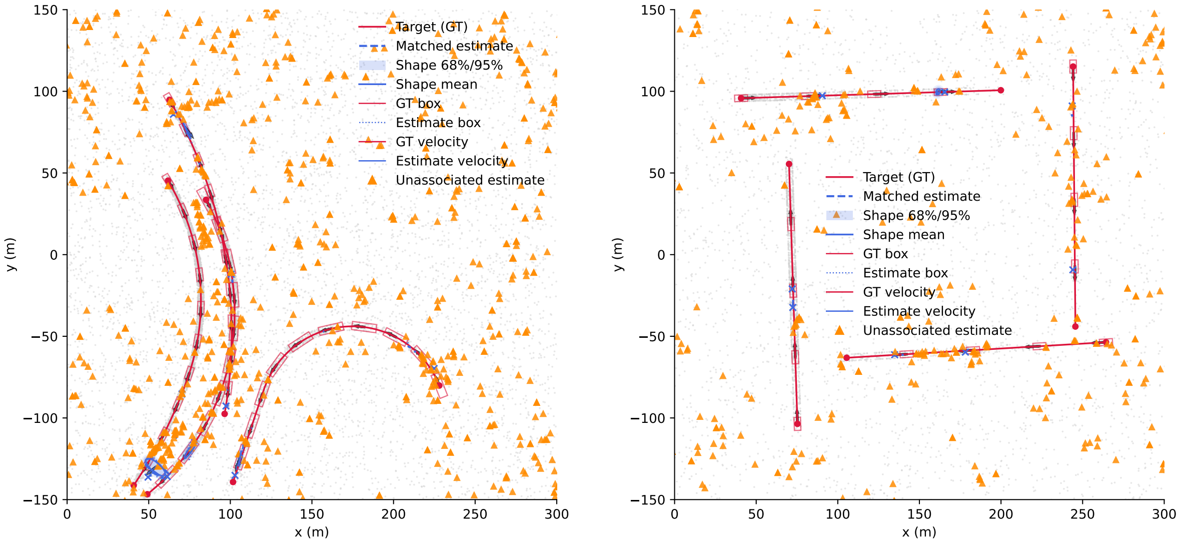
**(a)** Experiment: Waterbus 2 (3D)

**(b)** Experiment: Waterbus 3 (2D)

**(c)** Simulated: 3 targets, complex motions

**(d)** Simulated: 3 targets, varying sizes

**Figure H.17:** PHD (PAKF) estimated tracks — track-by-detection results (2/3). Each subfigure shows estimates for the named experiment or simulation.



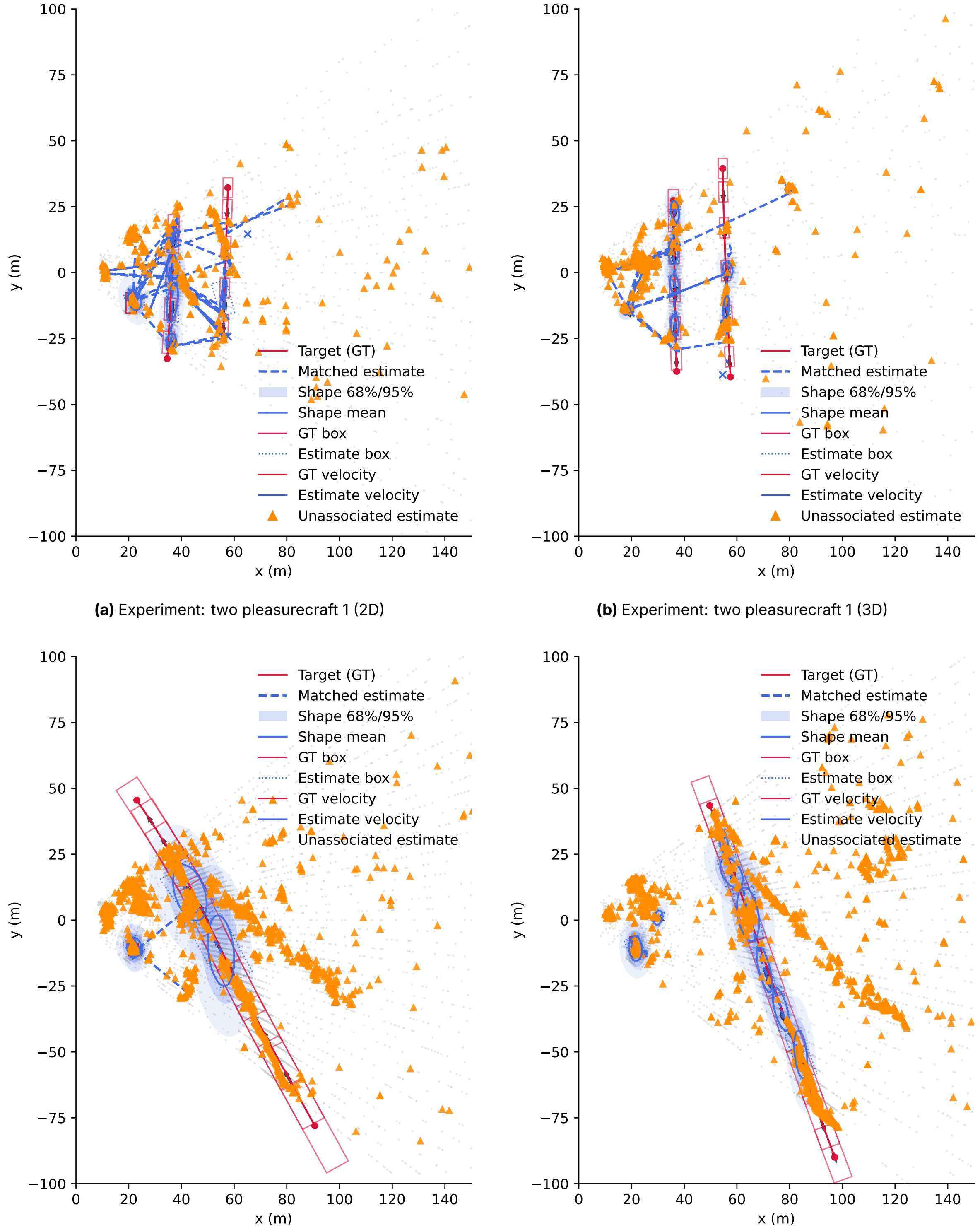
**(a)** Simulated: 4 targets, complex motions

**(b)** Simulated: 4 targets, rectangle CV

**Figure H.18:** PHD (PAKF) estimated tracks — track-by-detection results (3/3). Remaining subfigures for the named simulations.

## H.3. Poisson Multi-Bernoulli Mixture Filter

Only run for Gaussian Inverse Wishart



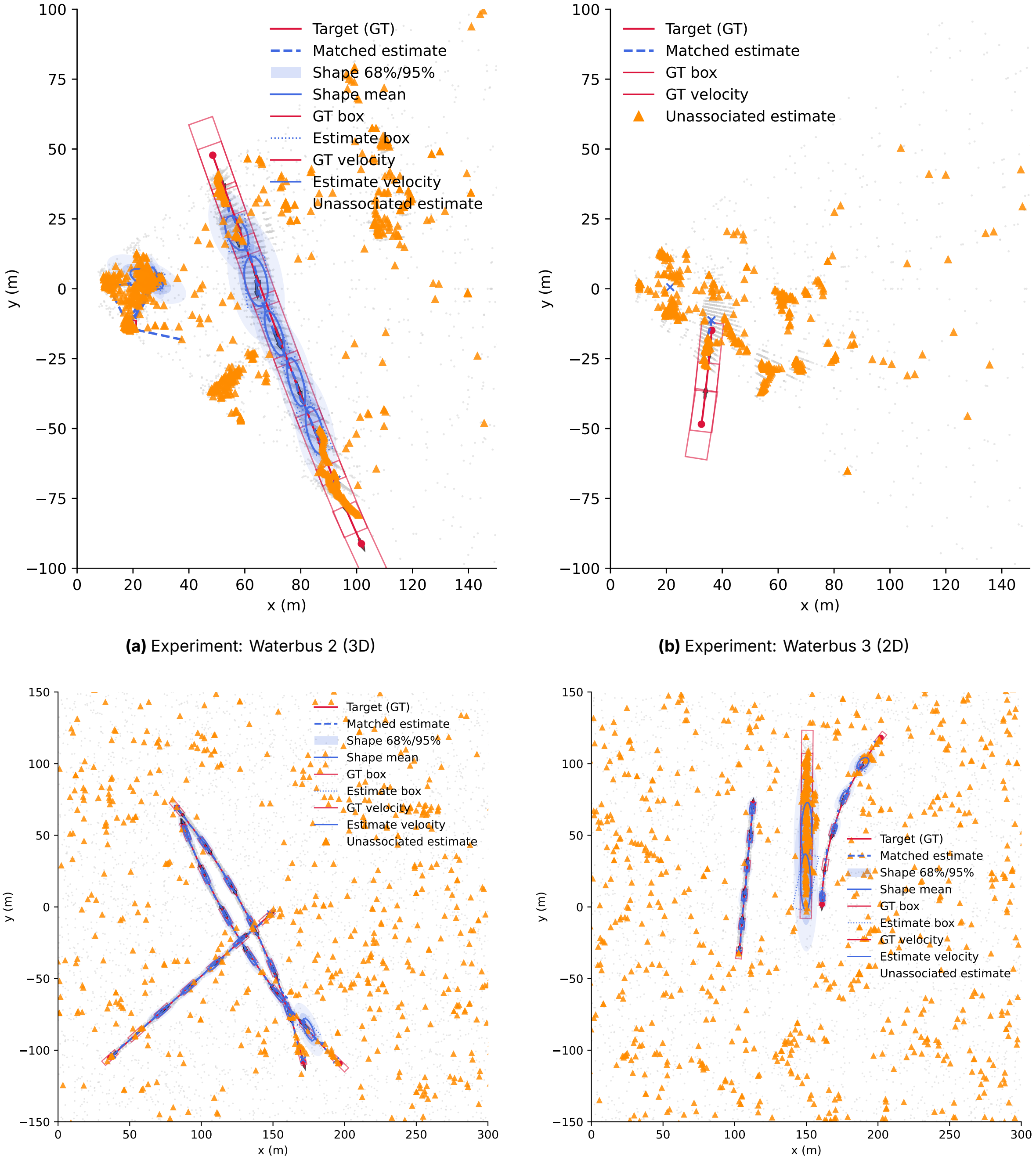
**(a)** Experiment: two pleasurecraft 1 (2D)

**(b)** Experiment: two pleasurecraft 1 (3D)

**(c)** Experiment: Waterbus 1 (2D)

**(d)** Experiment: Waterbus 2 (2D)

**Figure H.19:** PMBM estimated tracks — track-by-detection results (1/3). Each subfigure shows estimates for the named experiment or simulation.



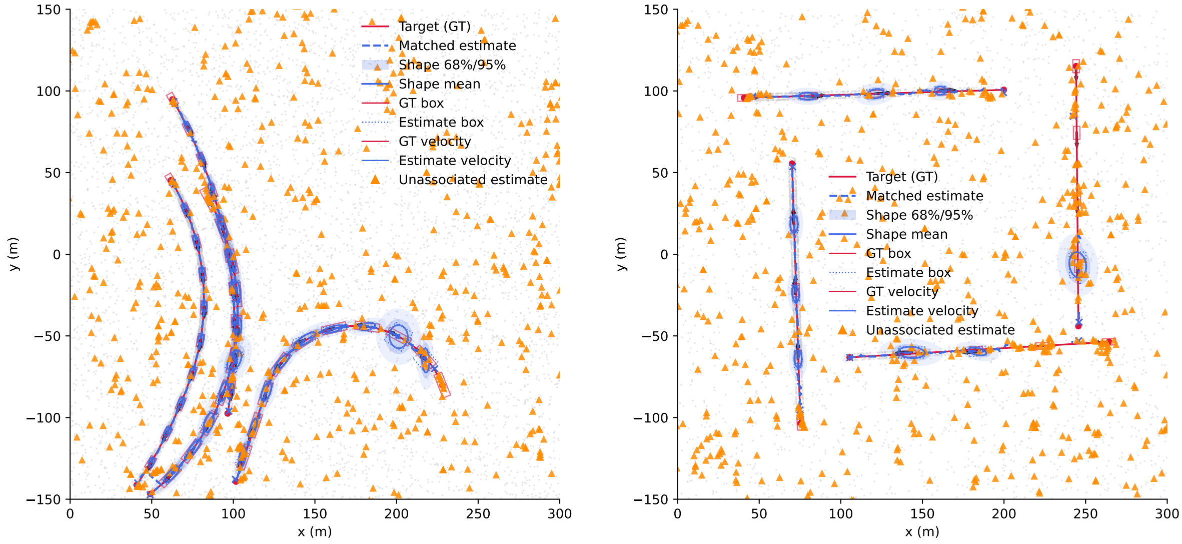
**(a)** Experiment: Waterbus 2 (3D)

**(b)** Experiment: Waterbus 3 (2D)

**(c)** Simulated: 3 targets, complex motions

**(d)** Simulated: 3 targets, varying sizes

**Figure H.20:** PMBM estimated tracks — track-by-detection results (2/3). Each subfigure shows estimates for the named experiment or simulation.



**(a)** Simulated: 4 targets, complex motions

**(b)** Simulated: 4 targets, rectangle CV

**Figure H.21:** PMBM estimated tracks — track-by-detection results (3/3). Remaining subfigures for the named simulations.