FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of master's thesis

Title:	Track coalescence in JPDA filters and its suppression
Student:	Bc. Artem Tokarevskikh
Supervisor:	doc. Ing. Kamil Dedecius, Ph.D.
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2024/2025

Instructions

The joint probabilistic data association (JPDA) filters represent a widely employed class of multi-target tracking algorithms, particularly favored in scenarios characterized by low clutter rates. Despite the availability of more modern random finite set (RFS)-based solutions such as the PHD or PMBM filters, the JPDA filters remain popular due to their lower computational demands. However, a notable drawback of JPDA is the track coalescence effect, wherein estimated tracks tend to merge when targets move in close proximity. This happens, e.g., if the moving targets cross or if they move in parallel. A formation of moving targets is then an extreme example. This thesis focuses on studying this coalescence effect and explores techniques for its suppression.

Goals:
1) Study the PDA filter and its multi-target variants, e.g., the JPDA and/or JIPDA filters.
2) Study the track coalescence problem. Discuss possible strategies for its suppression.
3) Propose a coalescence suppression method. Perform its assessment on simulated data and evaluate obtained results.
4) Discuss the results and indicate possible future research directions.

References:
[1] Kropfreiter, T., Meyer, F., Crouse, D. F., Coraluppi, S., Hlawatsch, F., & Willett, P. (2023). To Coalesce or to Repel? An Analysis of MHT, JPDA, and Belief Propagation Multitarget Tracking Methods. ArXiv. /abs/2308.06326
[2] E. Brekke, Fundamentals of Sensor Fusion. NTNU, 2020.

Electronically approved by Ing. Magda Friedjungová, Ph.D. on 7 February 2024 in Prague.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Master's thesis

Track coalescence in JPDA filters and its suppression

Bc. Artem Tokarevskikh

Department of Applied Mathematics

Supervisor: doc. Ing. Kamil Dedecius, Ph.D.

May 9, 2024

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60 (1) of the Act.

In Prague on May 9, 2024 . . . .

Czech Technical University in Prague

Faculty of Information Technology

Citation of this thesis

Tokarevskikh, Artem. *Track coalescence in JPDA filters and its suppression.* Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2024.

Abstrakt

Multi-target tracking (MTT) hraje klíčovou roli v různých aplikacích, od autonomních vozidel a robotiky až po dohled a řízení letového provozu. Filtry Joint Probabilistic Data Association (JPDA) se pro MTT široce používají díky své efektivitě a schopnosti zpracovávat nejistoty při asociaci dat v zašuměném prostředí. Nicméně filtry JPDA jsou náchylné ke splývání stop, kdy se stopy blízkých cílů mají tendenci slučovat, což vede k nepřesnému odhadu stavu a potenciální ztrátě stopy.

Tato diplomová práce zkoumá problém splývání stop ve filtrech JPDA a navrhuje novou techniku potlačení, která kombinuje adaptivní validační gateování s prořezáváním hypotéz. Naše metoda dynamicky upravuje velikost validační oblasti na základě blízkosti cílů a odhadovaných stavů, čímž efektivně omezuje možnosti asociace a snižuje pravděpodobnost splývání stop. Prostřednictvím simulací a experimentů demonstrujeme efektivitu našeho navrhovaného přístupu při zlepšování přesnosti lokalizace a celkového výkonu sledování ve srovnání se základním filtrem JPDA a JPDA* s prořezáváním.

Ačkoli naše metoda vykazuje významné výhody, vykazuje vyšší míru chybného detekování v extrémních scénářích. Diskutujeme potenciální vysvětlení tohoto omezení a navrhujeme směry pro budoucí výzkum, včetně adaptivního výběru parametrů gateování, alternativních strategií gateování a pokročilých technik správy stop. Řešením těchto aspektů se snažíme dále zlepšit robustnost a spolehlivost sledovacích systémů založených na JPDA pro zvládání komplexních scénářů s více cíli.

Klíčová slova Sledování více cílů, Joint Probabilistic Data Association (JPDA), splývání stop, adaptivní validační gateování, prořezávání hypotéz

Abstract

Multi-target tracking (MTT) plays a crucial role in various applications, from autonomous vehicles and robotics to surveillance and air traffic control. Joint Probabilistic Data Association (JPDA) filters have been widely employed for MTT due to their efficiency and ability to handle data association uncertainties in cluttered environments. However, JPDA filters are susceptible to track coalescence, where closely spaced target tracks tend to merge, leading to inaccurate state estimation and potential track loss.

This thesis investigates the problem of track coalescence in JPDA filters and proposes a novel suppression technique that combines adaptive validation gating with hypothesis pruning. Our method dynamically adjusts the validation gate size based on target proximity and estimated states, effectively limiting association possibilities and reducing the likelihood of track merging. Through simulations and experiments, we demonstrate the effectiveness of our proposed approach in improving localization accuracy and overall tracking performance compared to baseline JPDA and JPDA* with pruning.

While our method shows significant advantages, it exhibits a higher misdetection rate in extreme scenarios. We discuss potential explanations for this limitation and propose directions for future research, including adaptive gating parameter selection, alternative gating strategies, and advanced track management techniques. By addressing these aspects, we aim to further enhance the robustness and reliability of JPDA-based tracking systems for handling complex multi-target scenarios.

Keywords Multi-target tracking, Joint Probabilistic Data Association (JPDA), track coalescence, adaptive validation gating, hypothesis pruning

Contents

List of Figures

Introduction

Target tracking is the fundamental process of determining the location and trajectory of one or more moving objects over time using a series of measurements. It is a complex task that involves the ability to detect potential targets, separate true targets from false alarms or clutter, associate sequential measurements to the appropriate targets, and then estimate the target's state. Target states typically include kinematic components such as position and velocity and may encompass other attributes such as size, shape, or classification.

Target tracking finds widespread and often critical applications in a multitude of domains:

- **Microscopic Applications** [1], [2] In microbiology and other scientific fields, target tracking is used to study the movement and behavior of cells, bacteria, and other microscopic organisms. This type of tracking provides crucial information for research on diseases, cell function, and biological processes. [3] Self-driving vehicles and various robotic systems depend on target tracking for navigation, obstacle avoidance, and decision making. The need to perceive and track dynamic elements in the environment is a core requirement for autonomy.

- **Wildlife Monitoring** [4] Target tracking is used in ecological studies and wildlife preservation. Researchers use it to track animal movement patterns, migration routes, and population densities. These data help them understand animal behavior, habitat selection, and support conservation efforts.

- **Aerospace and Defense** [5] Essential for air traffic control, missile guidance systems, early warning systems, and threat assessment within military operations. Tracking systems are crucial for the safe and efficient navigation of aircrafts and to protect assets from potential danger.

- **Surveillance** [6] systems rely on target tracking for perimeter security, crowd monitoring, incident detection, and crime prevention. The ability to track individuals or vehicles plays an important role in ensuring safety in both public and private settings.

Filtering algorithms form the foundation of target tracking methodologies. The word *filter* in this context originates from its use in signal processing and electrical engineering. In these fields, a filter is a system or process that selectively modifies, removes, or enhances specific aspects of a signal. Early tracking algorithms, particularly the Kalman Filter [7], were built on the concepts of extracting meaningful information (in this case, the true state of the target) from noisy and imperfect data. This mirrors the action of an electrical filter that can separate a desired frequency range from a complex signal.

Filters are algorithms that dynamically estimate the state of a target in the presence of uncertainty and noise. These filters work by combining imperfect sensor measurements with a mathematical model that describes the expected motion patterns of the target. Here is a brief overview of some of the most widely used filters in target tracking.

- **Kalman Filter (KF)** The Kalman Filter [7] is the cornerstone of filtering techniques. It is designed for linear systems and Gaussian noise distributions. The KF works by recursively updating the target state estimate and its associated uncertainty.

- **Extended Kalman Filter (EKF)** Designed for non-linear systems. The EKF linearizes the non-linear system model around the current state estimate. This linearization allows us to apply the principles of the Kalman Filter to the non-linear system.

- **Particle Filter (PF)** Particle filters [8] excel in handling non-linear and non-Gaussian scenarios. They approximate the probability distribution of the target state using a set of weighted samples (particles). PFs are computationally more intensive, but their flexibility in modeling complex scenarios makes them invaluable for certain tracking applications.

- **Unscented Kalman Filter (UKF)** UKF [9] employs deterministic sampling to handle non-linearities, potentially offering improved accuracy over the EKF.

The choice of a particular filter largely depends on the characteristics of the system being tracked (for example, its dynamics and noise models) and the computational resources available for the task.

Target tracking poses numerous challenges that modern algorithms seek to address. They must deal with **noisy measurements** that introduce uncertainty in the estimation of the state of a target. **Errors in sensor readings** of a target's position, velocity, or other attributes can propagate and degrade the accuracy of the track. Furthermore, the tracking task is made more difficult by the presence of **clutter** in the sensor data. Clutter can originate from background noise, reflections, natural phenomena, or deliberate countermeasures, leading to **false detections** that obfuscate the identification of true targets. Furthermore, no sensor is perfect, even the most advanced detectors can occasionally miss true targets (resulting in false negatives) or mistake a clutter source for an actual target (introducing false positives). Such misdetections can create confusion and disrupt the continuity of a target's established track.

These challenges underscore the importance of designing robust target tracking techniques. Robustness in this context means resilience - the ability of the tracking system to maintain reliability and generate accurate estimates, even when faced with the inevitable imperfections of noise, clutter, and potential misdetections.

While the Kalman Filter and its variants provide a foundation for target tracking, they can be susceptible to disruption under challenging conditions. Although they handle noisy measurements and occasional misdetections, clutter represents a significant source of failure. To tackle single-target tracking in the presence of clutter, algorithms like the following are employed:

- **Nearest Neighbor (NN)** [10] Chooses the best data association and prunes all others in a greedy manner to approximate the posterior. This has the disadvantage of underestimating the posterior uncertainty.
- **Probabilistic Data Association (PDA)** [11] Instead of selecting only the most probable association hypothesis, the PDA filter merges all valid hypotheses into the posterior weighted by their likelihoods. This approach acknowledges uncertainties better than NN.
- **Multiple Hypothesis Tracking (MHT)** [12] Maintains multiple potential track hypotheses over time, reducing the risk of track loss due to misassociations.

For the more complex problem of multi-target tracking, two major families of robust tracking algorithms exist:

- **Associative Filters** These filters rely on establishing associations between measurements and existing tracks. Notable examples include **MHT**, **Joint Probabilistic Data Association (JPDA)** [13], and **Global Nearest Neighbor (GNN)** approaches.

- **Random Finite Set (RFS)** based filters. A different mathematical framework that models the set of targets directly. Examples include **Probability Hypothesis Density (PHD)** [14].

While the power and sophistication of RFS based filters offer advantages in complex scenarios, their in-depth discussion falls outside the scope of this thesis.

A significant challenge unique to multi-target tracking, particularly in the context of associative filters, is the potential for track coalescence. This phenomenon occurs when multiple targets move in close proximity, causing their measurements to overlap and become difficult to distinguish. The filter may converge onto a single track, as shown in Figure 0.1, representing the center of the group rather than maintaining distinct tracks for each individual target.

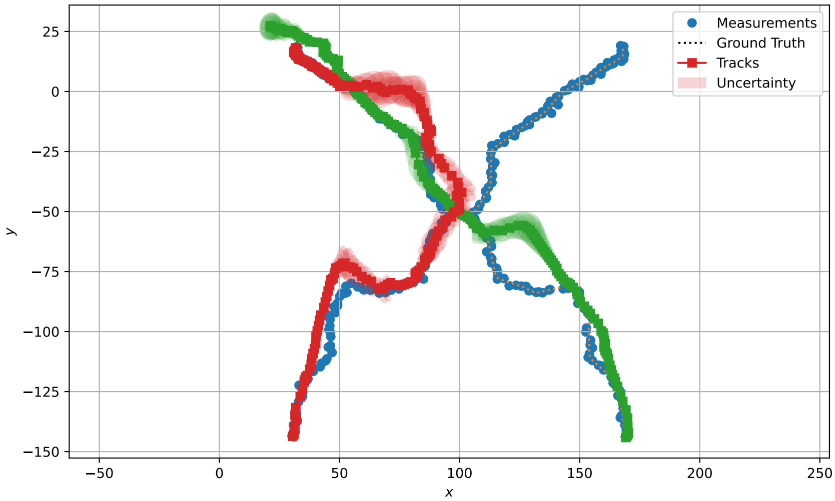


Figure 0.1: Coalescence example.

Track coalescence disrupts accurate count and state estimation of targets, which is crucial for applications that require situational awareness. It becomes progressively more pronounced in scenarios with dense clutter and as the number of closely spaced targets increases.

In this thesis, we focus on suppressing the coalescence effect in JPDA filters. This effect is widely recognized, and numerous diverse techniques have been developed for its mitigation. The most common are the exact nearest-neighbor PDA (ENNPDA) [15], its extensions JPDA* and CPDA* [16] based on hypothesis pruning. Some methods make use of the concept of a joint state for multiple targets. For example, JPDAC [17] or coupled PDA (CPDA) introduced in [16].

In the following chapters, we will dive into the aspects of single-target and multi-target tracking. In Chapter 3 we discuss the coalescence suppression approaches specifically for JPDA and present our method. Chapter 4 describes the experimental validation of the suggested method.

CHAPTER 1

Single target tracking

This chapter is dedicated to the topic of single-target tracking (STT). STT is a critical component in the field of computer vision and radar systems, focusing on the continuous observation and prediction of a single object's position and trajectory over time. This process involves detecting the object in the initial frame and then estimating its location in subsequent frames, despite any potential changes in appearance, occlusions, or motion.

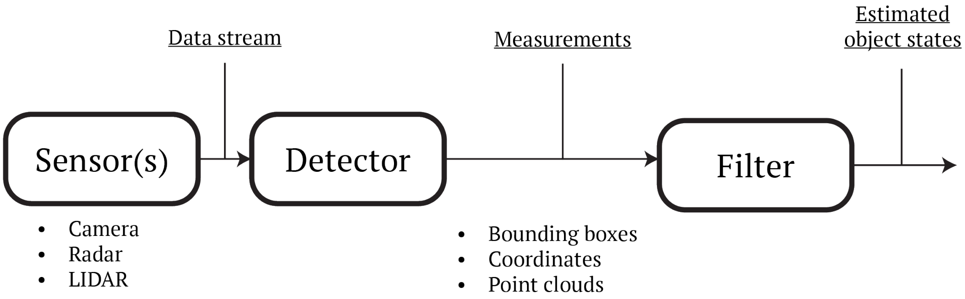


Figure 1.1: Tracking scheme.

In target tracking, the goal is to estimate the state of the object based on previous measurements, as shown in Figure 1.1. The state of an object is presented by the **state vector**.

Definition 1 (State vector) *Let X be the column vector that contains the set of parameters describing a system, known as states. We refer to this vector as the state vector.*

For example, the goal is to track an object in freefall along some axis (say x), as shown in Figure 1.2. The state vector for this case could have the form

$X = [x,\ z,\ v_x,\ v_z]^\intercal$, where x and z are the positions of an object and v_x, v_z are the corresponding velocities along these axes.

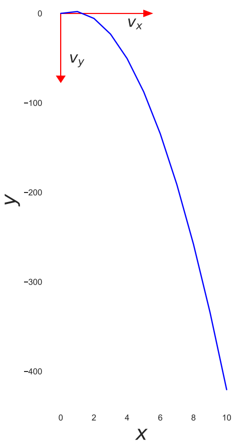


Figure 1.2: Falling object trajectory.

The state of an object at the time scan t is expressed as follows.

$$X_t = F_t X_{t-1} + v_t, \qquad v_t \sim \mathcal{N}(0, Q) \tag{1.1}$$

where F_t is the state transition matrix (motion model). The term v_t represents process noise, a zero-mean Gaussian term with covariance matrix Q. This noise captures deviations from the state transition model F_t. These deviations can arise not only from external or internal process disturbances, but also from real-world complexities that are often omitted for the simplicity of the model.

Due to practical limitations, the complete state of an object, denoted by X_t, may not be fully measurable. Often, only a subset of state variables can be directly observed through sensors. This observed subset, denoted by Z_t, represents the measurement vector. However, the measurement process itself is not perfect and introduces inaccuracies. These inaccuracies arise from sensor limitations, environmental noise, and other factors. Z_t is expressed as

$$Z_t = H_t X_t + w_t, \qquad w_t \sim \mathcal{N}(0, R) \tag{1.2}$$

where H_t is a simple matrix (measurement model) that selects the state variables observed in the measurement and w_t represents the uncertainty of the measurement. Returning to the example with falling object, let us say that we are only able to observe the coordinates. Then, the matrix H is time invariant and

$$\begin{aligned}
H &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \\
Z_t &= H X_t + w_t \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_t \\ z_t \\ v_{x_t} \\ v_{z_t} \end{bmatrix} + w_t \\
&= \begin{bmatrix} x_t \\ z_t \end{bmatrix} + w_t
\end{aligned}$$

Choosing a matrix F_t is equivalent to choosing a motion model for the process to model. We will describe it in the next section.

1.1 Motion model

A motion model is a mathematical representation that describes how a target's position and other state variables (such as velocity and acceleration) change over time. It is used to predict the future state of a target based on previous states, thereby enabling the tracking system to maintain a continuous estimation of the target's trajectory.

The choice of an appropriate motion model is crucial as it directly influences the filter's ability to track targets with different motion patterns. We will go through the most common ones.

Constant velocity model

The constant velocity model (CVM) is a simple linear motion model in which a tracked object travels at a constant speed. In fact, this model can be used even in cases where the speed varies over time without drastic changes. In this way, we are "hiding" some physical properties of the modeled process behind the noise term for the sake of simplicity, as we mentioned earlier.

Let us say that we track an object in 2D space with axes x and y. We also model the velocities of this object along the corresponding axes as a random walk. That is,

$$\begin{aligned} x_t &= x_{t-1} + v_{x_{t-1}}\Delta_t + w_{x_t}, \\ y_t &= y_{t-1} + v_{y_{t-1}}\Delta_t + w_{y_t}, \\ v_{x_t} &= v_{x_{t-1}} + w_{v_{x_t}}, \\ v_{y_t} &= v_{y_{t-1}} + w_{v_{y_t}}. \end{aligned}$$

Having the state vector $X = [x\ y\ v_x\ v_y]^\intercal$, we can rewrite our process in matrix form as follows

$$X_t = F_t X_{t-1} + v_t \tag{1.3}$$

$$\begin{bmatrix} x_t \\ y_t \\ v_{x_t} \\ v_{y_t} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta_t & 0 \\ 0 & 1 & 0 & \Delta_t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ v_{x_{t-1}} \\ v_{y_{t-1}} \end{bmatrix} + \begin{bmatrix} w_{x_t} \\ w_{y_t} \\ w_{v_{x_t}} \\ w_{v_{y_t}} \end{bmatrix}. \tag{1.4}$$

The simplicity of CVM makes it a valuable starting point for tracking systems. Although it may oversimplify the motion of real-world objects subject to acceleration or maneuvers, the CVM remains a widely used and often effective model, especially in scenarios involving short tracking periods or relatively stable target motion.

Constant acceleration model

The Constant Acceleration Model (CAM) builds upon the CVM by incorporating a constant acceleration component. This model is suitable for tracking objects that experience consistent changes in their velocity, such as a vehicle accelerating along a straight path or an object in freefall.

In the CAM, we extend the state vector to include acceleration components along the x and y axes, that is, $X = [x \; y \; v_x \; v_y \; a_x \; a_y]^\intercal$.

The process can then be modeled in matrix form as:

$$X_t = F_t X_{t-1} + v_t \tag{1.5}$$

$$\begin{bmatrix} x_t \\ y_t \\ v_{x_t} \\ v_{y_t} \\ a_{x_t} \\ a_{y_t} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta_t & 0 & \frac{1}{2}\Delta_t^2 & 0 \\ 0 & 1 & 0 & \Delta_t & 0 & \frac{1}{2}\Delta_t^2 \\ 0 & 0 & 1 & 0 & \Delta_t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta_t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ v_{x_{t-1}} \\ v_{y_{t-1}} \\ a_{x_{t-1}} \\ a_{y_{t-1}} \end{bmatrix} + \begin{bmatrix} w_{x_t} \\ w_{y_t} \\ w_{v_{x_t}} \\ w_{v_{y_t}} \\ w_{a_{x_t}} \\ w_{a_{y_t}} \end{bmatrix}. \tag{1.6}$$

The CAM offers increased fidelity in modeling object motion compared to the CVM by accounting for acceleration. It is better suited for tracking scenarios where objects undergo more dynamic motion patterns. However, the CAM does introduce additional complexity due to the increased dimensionality of the state space. The choice between CVM and CAM depends on the expected motion patterns of the tracked objects and the desired trade-off between accuracy and computational complexity.

1.2 Challenges in target tracking

The accuracy and robustness of target tracking systems are continually challenged by a confluence of factors that can severely degrade performance. Among these are the issues of clutter, missed detections, and ambiguities in data association.

Clutter

One of the most challenging obstacles in target tracking lies in the presence of clutter. Clutter encompasses a wide range of misleading measurements or signals that can significantly compromise the ability to reliably identify and track true targets.

In essence, clutter arises from any sensor detection that does not correspond to the actual object being tracked. The origins of clutter are numerous and diverse. Here are some of the most common sources.

- **Natural Environment** The natural world is a major contributor to clutter. Reflections of the ground and sea, weather conditions (rain, snow, fog), vegetation, and even the movement of wildlife or insects can generate false returns that interfere with target detection.
- **Human-Made Environments** Urban areas and other human-modified landscapes often generate substantial clutter. Buildings, bridges, power lines, and other structures can reflect or obscure sensor signals. Additionally, the presence of other vehicles, machinery, or unintentional emissions from electronic devices can create confusing measurements.
- **Sensor Limitations** Imperfections inherent in the sensor systems themselves lead to clutter. Random electronic noise, sidelobes in radar antennas, or limitations in resolution can introduce spurious detections.

It can significantly disrupt the processes of data association and state estimation within tracking algorithms.

Say, we are tracking some object in some area with volume V, which we split into N resolution cells. If we assume that the detection processes in different resolution cells are i.i.d., then the total number of clutter measurements, which we denote ϕ, is distributed according to the binomial distribution [18]

$$\mu(\phi) = \binom{N}{\phi} P_{clutter}^{\phi}(1 - P_{clutter})^{N-\phi}, \tag{1.7}$$

where $P_{clutter}$ is the probability of clutter occurrence. When N is large, the binomial distribution can be approximated with a Gaussian or Poisson distribution. The Poisson approximation is a standard model for clutter in target tracking. Leaving $\Lambda = NP_{clutter}$, the Poisson clutter model becomes [18]

$$\mu(\phi) = e^{-\Lambda}\frac{\Lambda^{\phi}}{\phi!}. \tag{1.8}$$

In Figure 1.3 we can see the radar simulations for different values of Λ. The Poisson distribution (1.8) is related to a more complicated construction, the

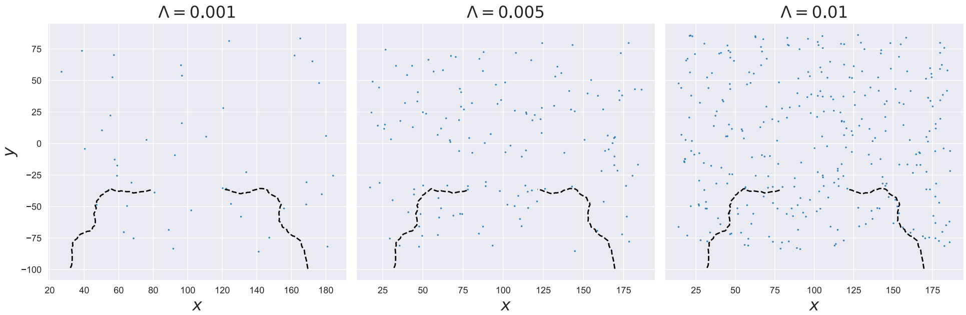


Figure 1.3: Measurements received at time $t = 50$ for different Λ. Dashed lines represent the tracked object trajectory in previous steps.

Poisson point process (PPP). From PPP we first sample a random number of points according to (1.8), and then sample the locations of these points according to a uniform pdf in the tracked region with volume V[18]. This gives us the framework for modeling clutter for simulations.

Misdetections

The second main challenge in target tracking is that the target may or may not be detected. In order to reflect this in modeling process we introduce the detection model, given by Bernoulli random variable:

$$P(\delta) = \begin{cases} P_D & \text{if } \delta = 1 \\ 1 - P_D & \text{if } \delta = 0 \end{cases}, \tag{1.9}$$

where P_D is probability of detection. The main complication is to accurately estimate this probability. Usually it is done via domain knowledge, e.g., 0.5 for sonar tracking applications or 0.95 for airplane radar tracking [18]. If P_D is too high, we increase the probability of reporting a false alarm when the target is not present. Selecting an appropriate trade-off is a critical aspect of the modeling process.

Unknown data associations

In target tracking, data association, also known as correspondence problem, refers to the critical task of determining which new measurements collected by the sensor originate from the targets currently being tracked as shown in Figure 1.4. This process becomes particularly complex in multi-target scenarios where multiple potential measurement-to-track pairings exist. The presence of clutter significantly complicates this challenge, as the tracking system must now also discriminate between genuine target detections and spurious measurements.

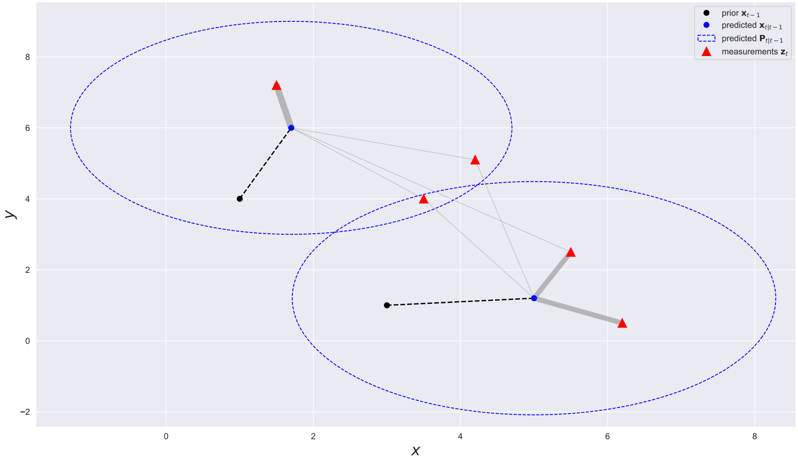


Figure 1.4: All possible data association for two tracks and 5 received measurements. Line thickness is proportional to likelihood of measurement being produced by the target.

Clutter introduces a high degree of uncertainty into the data association process. False measurements compete with valid detections generated by true targets. Incorrectly assigning a clutter measurement to an existing track can lead to track contamination, where erroneous information corrupts the target state estimate. On the contrary, mistaking a valid target detection for clutter can result in missed detections and potential track loss.

To address the challenges of data association in cluttered environments, various techniques have been developed.

- **Validation Gating** A preliminary filtering step to exclude measurements that are highly unlikely to have originated from the tracked targets based on their predicted positions and uncertainties. This reduces the number of potential associations that need to be considered. The validation gate is depicted in Figure 1.4 as a blue dashed ellipse.
- **Nearest Neighbor (NN)** A classic approach that chooses the one optimal hypothesis according to selected criterion.
- **Probabilistic Data Association (PDA)** This framework takes a probabilistic approach by assigning weights (probabilities) to different measurement-to-track association hypotheses. This allows the filter to account for uncertainties arising from clutter and the possibility of misdetections. We will dive deep into the PDA algorithm in Section 1.5.

- **Multiple Hypothesis Tracking (MHT)** A more computationally intensive approach that maintains multiple hypotheses for data association over time. This allows for the possibility that initial association decisions might be revised with the arrival of new information. This method is beyond the scope of this thesis.

The choice of data association technique depends on factors such as the target density, the expected clutter density, and the available computational resources. Effective data association in the presence of clutter is a crucial component of robust target tracking systems, especially within complex multi-target environments.

The challenges inherent in target tracking, including clutter, misdetections, and data association uncertainties, highlight the vital need for robust filtering algorithms. Let us dive into the filtering algorithms and start with the most simple Bayes filter. We will present a theoretical background and basic principles applied in filtering algorithms.

1.3 Bayesian filtering

In the context of the filtering problem, our goal is to estimate the state of a stochastic dynamic system based on a sequence of noisy measurements. The state of the system will be denoted by $\mathbf{x}_t$.

For the filtering algorithm we do need motion and measurement models, as stated earlier. The motion model deals with how the state evolves with time based on the previous state. Mathematically, it can be expressed as $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. The measurement model concerns on how the measurements $\mathbf{z}_t$ we observe are related to the non-observable state $\mathbf{x}_t$. This model specified as $p(\mathbf{z}_t|\mathbf{x}_t)$.

A couple of independence assumptions underlie this way of treating the filtering problem. For the given specification of the measurement model to make sense, the Markov property must hold [18]

$$p(\mathbf{x}_t|\mathbf{x}_1,\ldots,\mathbf{x}_{t-1},\mathbf{z}_1,\ldots,\mathbf{z}_{t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}). \tag{1.10}$$

This means that the state $\mathbf{x}_t$ depends solely on $\mathbf{x}_{t-1}$. Similar assumption is assumed to hold for measurements

$$p(\mathbf{z}_t|\mathbf{x}_1,\ldots,\mathbf{x}_{t-1},\mathbf{x}_t,\mathbf{z}_1,\ldots,\mathbf{z}_{t-1}) = p(\mathbf{z}_t|\mathbf{x}_t). \tag{1.11}$$

The goal of the filtering problem in the Bayesian approach is to find the posterior distribution $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ [18]. Once the posterior distribution is specified, we are able to calculate the estimates of $\mathbf{x}_t$, for example, the MMSE or MAP estimation.

Filtering is performed in two steps: prediction and update. Prediction step is based on the **Chapman-Kolmogorov equation** which is based on total probability theorem and is given by

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) &= \int p(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{z}_{1:k-1}) d\mathbf{x}_{t-1} \\ &= \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{z}_{1:k-1}) d\mathbf{x}_{t-1}. \end{aligned} \tag{1.12}$$

Update step utilizes the **Bayes rule**, which is defined as

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{z}_{1:t}) &= \frac{p(\mathbf{z}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{z}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})} \\ &\propto p(\mathbf{z}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{z}_{1:t-1}). \end{aligned} \tag{1.13}$$

Equations (1.12) and (1.13) together referred to as the Bayes filter and serve as the basis for all filtering algorithms that follow. However, these equations do not have a general closed-form solution and are not applicable as is. Furthermore, when we multiply two pdfs in Equation (1.13) we are not guaranteed that the normalized product will result in some known pdf [18].

1.4 Kalman filter

In 1960 Rudolf Emil Kálmán discovered [7] that under certain assumptions the closed-form solution to equations (1.12) and (1.13) exists. The algorithm he invented provides an optimal and recursive solution for estimating the state of a linear dynamic system from a series of noisy measurements, as shown in Figure 1.5. The algorithm is named after him. The assumptions under which the closed-form solution exists are the following.

1. **Linearity** The Kalman filter assumes that the underlying system dynamics and the relationship between the state and the measurements can be described by linear equations.

2. **Gaussian Noise** Process noise, which represents uncertainty in the target's motion model, and measurement noise are assumed to have Gaussian distributions with known covariance matrices.

The state vector $\mathbf{x}$ comprises the quantities of interest for the target, such as its position, velocity, and potentially acceleration, as in Equation (1.6). The

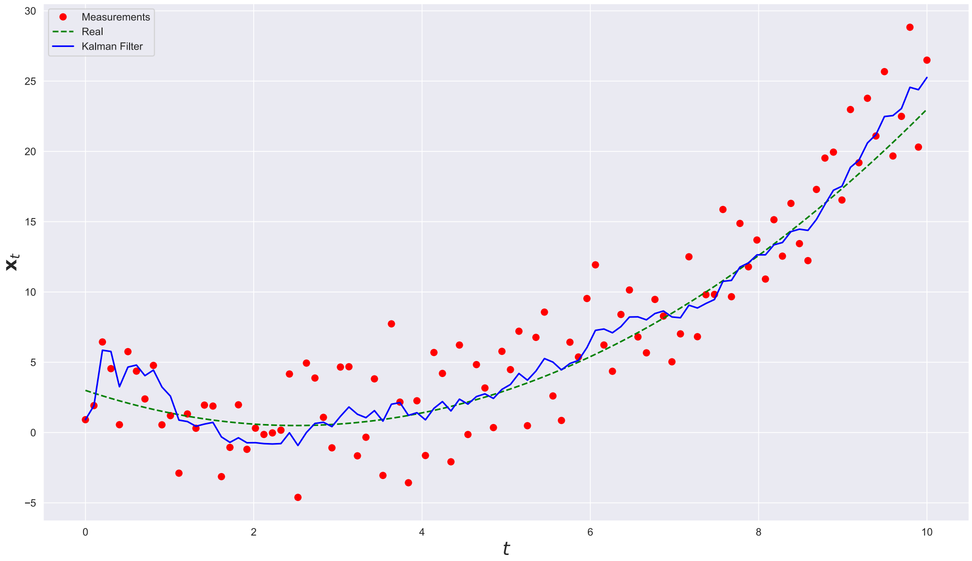


Figure 1.5: Kalman filter prediction for one dimensional state $\mathbf{x}_t$.

Kalman filter models the evolution of this state vector over time using a linear state transition Equation (1.1). Measurement model is given by Equation (1.2)

Given our understanding of both motion and measurement models, and our need for linearity and Gaussian likelihoods, we have

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \mathbf{F}\mathbf{x}_{t-1}, \mathbf{Q}), \tag{1.14}$$
$$p(\mathbf{z}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{z}_t; \mathbf{H}\mathbf{x}_t, \mathbf{R}), \tag{1.15}$$
$$p(\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0; \hat{\mathbf{x}}_0, \mathbf{P}_0), \tag{1.16}$$

where $\mathbf{F}$ and $\mathbf{H}$ are transition and measurement matrices of suitable dimensions, $\mathbf{Q}$ and $\mathbf{R}$ are corresponding positive semi-definite matrices, that govern the statistical properties of the process and measurement noise, and $\mathbf{P_0}$ is positive semi-definite matrix that expresses the prediction uncertainty. These equations are often written as

$$\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{P}_0), \tag{1.17}$$
$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{v}_t, \qquad \mathbf{v}_t \sim \mathcal{N}(0, \mathbf{Q}), \tag{1.18}$$
$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \mathbf{w}_t, \qquad \mathbf{w}_t \sim \mathcal{N}(0, \mathbf{R}). \tag{1.19}$$

The prediction step of Kalman filter algorithm

Algorithm 1 KF_Predict($\mathbf{x}_{t-1|t-1}, \mathbf{P}_{t-1|t-1}$)

```
    $\mathbf{x}_{t|t-1} \leftarrow \mathbf{F}\mathbf{x}_{t-1|t-1}$                    ▷ The predicted state estimate
    $\mathbf{P}_{t|t-1} \leftarrow \mathbf{F}\mathbf{P}_{t-1|t-1}\mathbf{F}^\intercal + \mathbf{Q}$      ▷ The predicted covariance
    $\mathbf{z}_{t|t-1} \leftarrow \mathbf{H}\mathbf{x}_{t|t-1}$                    ▷ The predicted measurement
    $\mathbf{S}_t \leftarrow \mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^\intercal + \mathbf{R}$          ▷ The measurement predicted covariance
```

In the prediction step, uncertainty $\mathbf{P}_{t|t-1}$ always increases since we make an assumption without new information. If, for some reason, the update step is skipped for subsequent time scans, the prediction uncertainty would increase fairly quickly. In the prediction step, we also predict the measurement and its covariance for further computation.

The pseudocode for the update step appears as follows.

Algorithm 2 KF_Update($\mathbf{x}_{t|t-1}, \mathbf{P}_{t|t-1}, \mathbf{z}_{t|t-1}, \mathbf{S}_t, \mathbf{z}_t$)

```
    $\nu_t \leftarrow \mathbf{z}_t - \mathbf{z}_{t|t-1}$                              ▷ The innovation
    $\mathbf{W}_t \leftarrow \mathbf{P}_{t|t-1}\mathbf{H}^\intercal\mathbf{S}_t^{-1}$                  ▷ The Kalman gain
    $\mathbf{x}_{t|t} \leftarrow \mathbf{x}_{t|t-1} + \mathbf{W}_t\nu_t$             ▷ The updated state estimate
    $\mathbf{P}_{t|t} \leftarrow (\mathbf{I} - \mathbf{W}_t\mathbf{H})\mathbf{P}_{t|t-1}$          ▷ The updated covariance
    return $\mathbf{x}_{t|t}$, $\mathbf{P}_{t|t}$,
```

In the update step, we introduce two new variables: *innovations* and the *Kalman gain*. Innovations are the difference between what we predict to measure and what we actually measure. We can think of them as the "new information" that our most recent measurement provides. If the innovation is large, it indicates that our prediction was significantly different from reality. It could mean either our model of the system is inaccurate or that there was some large, unexpected change in the system's dynamics.

Let us look closer to the Kalman gain equation.

$$\begin{aligned} \mathbf{W}_t &= \mathbf{P}_{t|t-1}\mathbf{H}^\intercal\mathbf{S}_t^{-1} && (1.20) \\ &= \mathbf{P}_{t|t-1}\mathbf{H}^\intercal(\mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^\intercal + \mathbf{R})^{-1} && (1.21) \\ &= \frac{\mathbf{P}_{t|t-1}\mathbf{H}^\intercal}{\mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^\intercal + \mathbf{R}} && (1.22) \end{aligned}$$

The value of Kalman Gain mainly depends on two values, measurement variance $\mathbf{R}$ and prediction uncertainty $\mathbf{P}_{t|t-1}$. Let us look at how the value of the

Kalman gain changes.

$$\lim_{\mathbf{R}\to 0} \frac{\mathbf{P}_{t|t-1}\mathbf{H}^\intercal}{\mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^\intercal + \mathbf{R}} = \mathbf{H}^{-1} \tag{1.23}$$

$$\lim_{\mathbf{P}_{t|t-1}\to 0} \frac{\mathbf{P}_{t|t-1}\mathbf{H}^\intercal}{\mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^\intercal + \mathbf{R}} = 0 \tag{1.24}$$

If we take the state update formula

$$\mathbf{x}_{t|t} = \mathbf{x}_{t|t-1} + \mathbf{W}_t(\mathbf{z}_t - \mathbf{z}_{t|t-1}) \tag{1.25}$$

we can see that when the magnitude of $\mathbf{P}_{t|t-1}$ is very low, which means that the state is predicted accuratly, we mostly discard the new measurements relying instead on the filter's prediction. On the other hand, when the measurements are accurate, and thus the magnitude of $\mathbf{R}$ is small, the updated state estimate relies mostly on the new measurements. So, Kalman gain is basically the measure of how much we want to change the predicted estimate by given a measurement.

The elegance of the Kalman filter and its computational efficiency have made it a fundamental building block for countless tracking applications. In the context of single-target tracking without clutter, the Kalman filter offers a principled framework for modeling target motion and refining state estimates as new sensor observations become available.

However, the Kalman filter is not able to handle clutter measurements. It cannot determine which of the measurements received are background clutter and which are the measurements generated by the target. For this purpose, we introduce the PDA filter.

1.5 Probability data association

PDA filter, similarly to the Kalman filter, operates under a set of assumptions. These assumptions are

1. **Single Target** At each time step, there exists at most one target in the surveillance region.
2. **Prior Density** The prior density of the target's state $\mathbf{x}_{t-1}$ is given as $p_{t-1}(\mathbf{x}_{t-1})$.
3. **Markov Model** The target's state evolves according to a Markov model $\mathbf{f}_\mathbf{x}(\mathbf{x}_t|\mathbf{x}_{t-1})$. This means that the state of the target at the current time step depends only on the previous time step's state, and not on any earlier states.

4. **Probability of Detection** The target detections occur independently over time with known probability P_D.

5. **Likelihood Function** If a measurement originates from the target, it is related to the target's state $\mathbf{x}_t$ through a known likelihood function $\mathbf{f}_{\mathbf{z}}(\mathbf{z}_t|\mathbf{x}_t)$.

6. **Clutter Distribution** The number of clutter measurements is unknown, but follows a PPP (1.8). In the context of the PDA filter, it represents the randomness in the number of clutter measurements received at each time step.

7. **Independent Clutter** A clutter measurement $\mathbf{z}$ is distributed according to a pdf $c(\mathbf{z})$, independently of all other measurements.

These assumptions allow the PDA filter to track a single target in the presence of clutter. However, it is important to note that these assumptions may not always hold true in real-world scenarios. For example, the probability of detection may vary depending on the target's range or aspect angle, and clutter measurements may exhibit some degree of spatial correlation.

At each time step, the filter gets the set of measurements $\mathbf{Z}_t = \{\mathbf{z}_t^{(1)}, \ldots, \mathbf{z}_t^{(m_t)}\}$ of variable length m_t. Note that the size of $\mathbf{Z}_t$ is unknown and varies over time. Figure 1.6 shows an example of filter receiving 3 measurements.

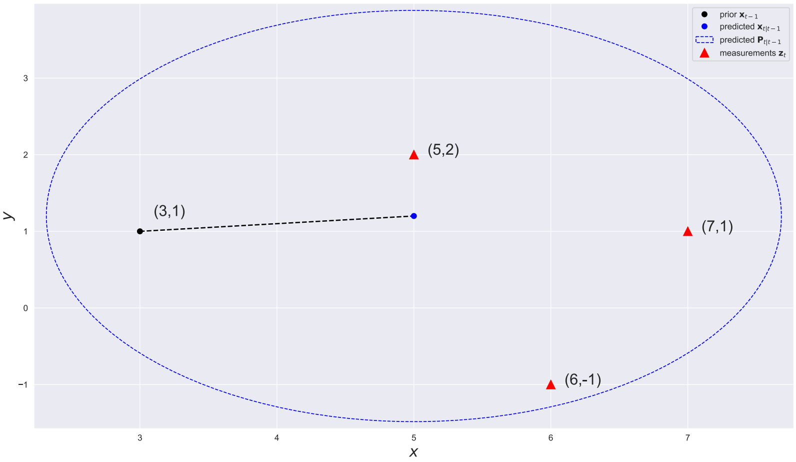


Figure 1.6: Example of $\mathbf{Z}_t$ of length 3.

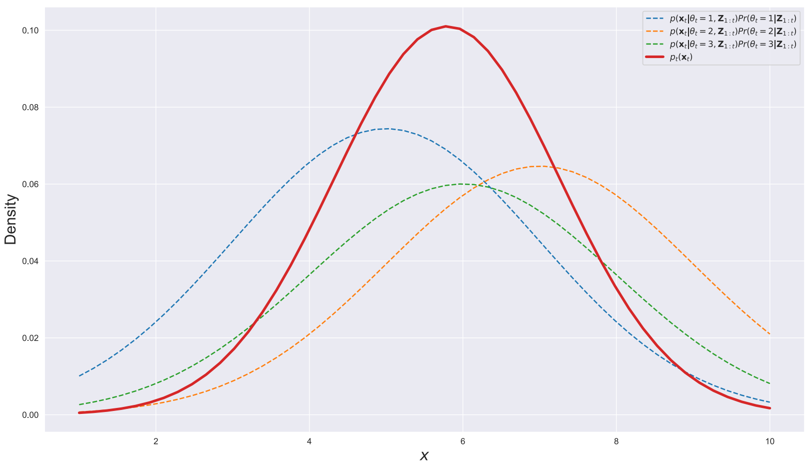


Figure 1.7: Mixture reduction for x coordinate of the state.

Assumption 1. is very convenient from a computational perspective, since instead of deciding which of several subsets of m_t is produced by the real target, we only need to determine which of the measurements received, if any, comes from the target. For this purpose, we introduce the association variable θ_t defined as

$$\theta_t = \begin{cases} 0 & \text{if target is undetected} \\ i \in \{1, \dots, m_t\} & \text{if } \mathbf{z}_{i,t} \text{ is produced by target} \end{cases} \tag{1.26}$$

According to the total probability theorem, the posterior density of the target state can be expressed as [18]

$$p_t(\mathbf{x}_t) = \sum_{i=0}^{m_t} p(\mathbf{x}_t|\theta_t = i, \mathbf{Z}_{1:t})Pr\{\theta_t = i|\mathbf{Z}_{1:t}\}. \tag{1.27}$$

We can clearly see that the posterior has the form of Gaussian mixture function. In order to obtain the posterior we need to perform the mixture reduction to a single Gaussian. The visual intuition behind the reduction of the mixture for one coordinate from the example above is presented in Figure 1.7. To do this, we need to specify the pdfs $p(\mathbf{x}_t|\theta_t = i, \mathbf{Z}_{1:t})$, and the event probabilities $Pr\{\theta_t = i|\mathbf{Z}_{1:t}\}$.

Let us start with the prediction step. It is based on the Chapman-Kolmogorov equation (1.12). Predicted pdf holds

$$p_{t|t-1}(\mathbf{x}_t) = \int \mathbf{f_x}(\mathbf{x}_t|\mathbf{x}_{t-1})p_t(\mathbf{x}_{t-1})d\mathbf{x}_{t-1}. \tag{1.28}$$

Before moving to update, recall that in order to get the closed-form solution, the motion model and the prior must be Gaussian. The motion model must also be linear. According to [18] the event-conditional posteriors $p(\mathbf{x}_t|\theta_t = i, \mathbf{Z}_{1:t})$ are given by

$$p(\mathbf{x}_t|\theta_t, \mathbf{Z}_{1:t}) \propto \begin{cases} p_{t|t-1}(\mathbf{x}_t) & \text{if } \theta_t = 0 \\ \mathbf{f_z}(\mathbf{z}_t^{(\theta_t)}|\mathbf{x}_t)p_{t|t-1}(\mathbf{x}_t) & \text{if } \theta_t > 0. \end{cases} \tag{1.29}$$

In the event that no measurement is associated with the target, the set $\mathbf{Z}_t$ provides no information about $\mathbf{x}_t$, so we use the predicted density. On the contrary, when $\theta_t > 0$ and we know the value of θ_t, then the state estimation boils down to the conventional Bayes rule, similarly as in the Kalman filter.

To find the event probabilities, let us first define the likelihood

$$l^{(\theta_t)} = \int \mathbf{f_z}(\mathbf{z}_t^{(\theta_t)}|\mathbf{x}_t)p_{k|k-1}(\mathbf{x}_t)d\mathbf{x}_t \tag{1.30}$$

Then, under the assumptions presented above, the posterior association probabilities are given by [18]

$$Pr\{\theta_t|\mathbf{Z}_{1:t}\} \propto \begin{cases} (1 - P_D)m_t\dfrac{\mu(m_t)}{\mu(m_t - 1)} & \text{if } \theta_t = 0 \\ \dfrac{P_D}{c(\mathbf{z}_t^{(\theta_t)})}l^{(\theta_t)} & \text{if } \theta_t > 0, \end{cases} \tag{1.31}$$

where $\mu(m_t)$ stands for Poisson clutter density defined in Equation (1.8).

Similarly as in Kalman filter section, let us define the continuous pdfs given the assumptions.

$$p(\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_{t-1}; \hat{\mathbf{x}}_{t-1}, \mathbf{P}_{t-1}), \tag{1.32}$$
$$\mathbf{f_x}(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \mathbf{F}\mathbf{x}_{t-1}, \mathbf{Q}), \tag{1.33}$$
$$\mathbf{f_z}(\mathbf{z}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{z}_t; \mathbf{H}\mathbf{x}_t, \mathbf{R}), \tag{1.34}$$
$$c(\mathbf{z}_t) = \frac{1}{V_t}, \tag{1.35}$$

where V_t is the volume of the surveillance area, in which measurements are considered. Given this linear Gaussian pdfs we get the classic Kalman prediction step where the predicted density is defined as

$$p_{t|t-1}(\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_t; \hat{\mathbf{x}}_{t|t-1}, \mathbf{P}_{t|t-1}), \tag{1.36}$$

and

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}\mathbf{x}_{t-1}, \tag{1.37}$$
$$\mathbf{P}_{t|t-1} = \mathbf{F}\mathbf{P}_{t-1}\mathbf{F}^{\intercal} + \mathbf{Q}. \tag{1.38}$$

The expressions for predicted measurements and covariance are identical to those in the Kalman filter, that is

$$\hat{\mathbf{z}}_{t|t-1} = \mathbf{H}\hat{\mathbf{x}}_{t|t-1}, \tag{1.39}$$

$$\mathbf{S}_t = \mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^\intercal + \mathbf{R}. \tag{1.40}$$

At this point, we obtain the expressions to calculate the posterior density of the state (1.27). For this, we have to compute both (1.29) and (1.31) for all m_t measurements. Essentially, we assume that we have 1 true measurement and $m_t - 1$ clutter measurements in the best-case scenario. Therefore, the $m_t - 1$ computational iterations are excessive and do not provide any information about the target's true state. In order to reduce computational complexity and eliminate improbable measurements, we use ellipsoidal gating, by adjusting the value of the V_t used in (1.35).

The name implies that the surveillance area is constrained by the ellipse, whose shape is defined by the predicted measurement (1.39) and its covariance (1.40) as in Figure 1.6. We also introduce the parameter γ, which defines the size of the validation region. The gating process is defined as follows.

$$\mathcal{V}(\mathbf{Z};\gamma) = \{\mathbf{z} \in \mathbf{Z} | (\mathbf{z} - \hat{\mathbf{z}}_{t|t-1})\mathbf{S}_t^{-1}(\mathbf{z} - \hat{\mathbf{z}}_{t|t-1}) < \gamma\}. \tag{1.41}$$

The parameter γ basically says how many standard deviations are we willing to consider. To meaningfully determine the value of γ, we introduce the probability of gating - P_G. Having this probability, we can find the parameter γ using the chi-square distribution with the degree of freedom equal to the dimensionality of the measurement vector [19]. The common strategy is to set P_G at some large probability, say $P_G = 0.995$ and then use the cumulative distribution of $\chi^2(n_\mathbf{z})$ to find γ, where $\mathbf{z}$ stands for the dimensionality of the measurement space. However, when we introduce the probability P_G we must then tweak our formulas to consider this probability. Basically, it boils down to replacing the term P_D with $P_G P_D$, for example, in (1.31). For simplicity, consider the probability of gating $P_G = 1$. Note that only measurements from $\mathcal{V}(\mathbf{Z};\gamma)$ are used in subsequent steps.

Let us now look at the update step. It is very simmilar to the Kalman filter. Firstly, the resulting posterior density is defined as [18]

$$p_t(\mathbf{x}_t|\theta_t, \mathbf{Z}_{1:t}) = \mathcal{N}(\mathbf{x}_t; \hat{\mathbf{x}}_t^{(\theta_t)}, \mathbf{P}_t^{(\theta_t)}), \tag{1.42}$$

where $\hat{\mathbf{x}}_t^{\theta_t}$, and $\mathbf{P}_t^{\theta_t}$ are given by the standard Kalman update. Let us see, how to obtain them.

Again, we calculate the Kalman gain using

$$\mathbf{W}_t = \mathbf{P}_{t|t-1}\mathbf{H}^\intercal\mathbf{S}^{-1}. \tag{1.43}$$

Following the computation of event-based innovations

$$\nu_t^{(\theta_t)} = \mathbf{z}_t^{(\theta_t)} - \mathbf{H}\hat{\mathbf{x}}_{t|t-1}. \tag{1.44}$$

Then the predicted event-conditioned estimate of $\hat{\mathbf{x}}_t^{(\theta_t)}$ is given by

$$\hat{\mathbf{x}}_t^{(\theta_t)} = \begin{cases} \hat{\mathbf{x}}_{t|t-1} & \text{if } \theta_t = 0 \\ \hat{\mathbf{x}}_{t|t-1} + \mathbf{W}_t \nu_t^{(\theta_t)} & \text{if } \theta_t > 0, \end{cases} \tag{1.45}$$

and its covariance

$$\mathbf{P}_t^{(\theta_t)} = \begin{cases} \mathbf{P}_{t|t-1} & \text{if } \theta_t = 0 \\ (\mathbf{I} - \mathbf{W}_t\mathbf{H})\mathbf{P}_{t|t-1} & \text{if } \theta_t > 0. \end{cases} \tag{1.46}$$

Nonetheless, these formulas take into account just a single measurement $\mathbf{z}_t^{(\theta_{i,t})}$, which gives us the mixture at the end of the filter iteration. In order to fulfill the above-stated Assumption 2., a reduction of the mixture is required. The new Gaussian must have the same expectation and covariance as the original mixture. This known as moment matching [18]. The mixture expectation is given by

$$\hat{\mathbf{x}}_t = \beta_t^{(0)}\hat{\mathbf{x}}_{t|t-1} + \sum_{\theta_t > 0} \beta_t^{(\theta_t)}\hat{\mathbf{x}}_t^{(\theta_t)}. \tag{1.47}$$

Let us define *combined innovations* as

$$\nu_t = \sum_{\theta_t > 0} \beta_t^{(\theta_t)} \nu_t^{(\theta_t)}. \tag{1.48}$$

This allows us to rewrite (1.47) as [18]

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t-1} + \mathbf{W}_t \nu_t. \tag{1.49}$$

Now, we need to calculate the corresponding data association probabilities $\beta^{(0)}$ and $\beta^{(\theta_t)}$. $\beta^{(0)}$ represents the probability that no gated measurements are produced by the target. The probability $\beta_i^{(\theta_t)}$ represents the probability that the tracked object produces the i-th measurement from $\mathcal{V}(\mathbf{Z};\gamma)$. Given the clutter density Λ we get

$$\beta^{(0)} = \frac{1}{c}\Lambda(1 - P_D P_G) \overset{P_G=1}{=} \frac{1}{c}\Lambda(1 - P_D) \tag{1.50}$$

$$\beta_i^{(\theta_t)} = \frac{1}{c} P_D l_{i,t}, \tag{1.51}$$

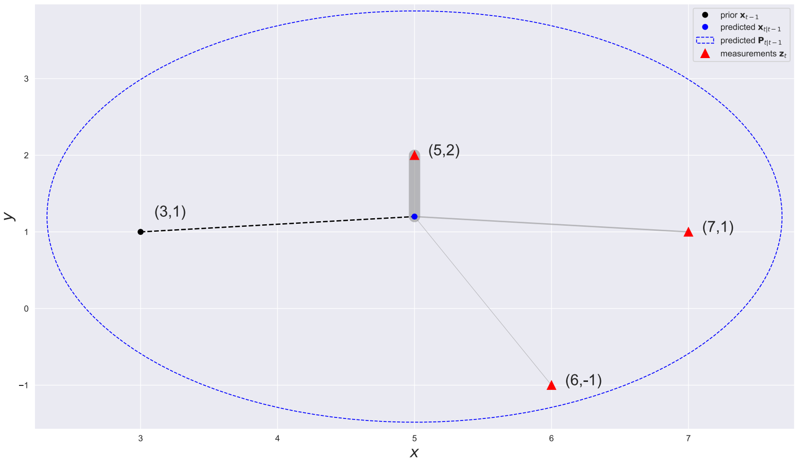


Figure 1.8: Measurement with association likelihoods. Grey lines' thickness represent the magnitude of the likelihood.

where c is normalization constant $c = \beta^{(0)} + \sum_i \beta_i^{(\theta_t)}$ and

$$l_{i,t} = \mathcal{N}(\mathbf{z}_{i,t}; \hat{\mathbf{z}}_{t|t-1}, \mathbf{S}_t) \tag{1.52}$$

is the measurement likelihood. Continuing the example in Figure 1.6, we depict the likelihoods of the measurements received in Figure 1.8.

Matching the second-order moments, meaning covariances, leads to [18]

$$\hat{\mathbf{P}}_t = \beta^{(0)} \mathbf{P}_{t|t-1} + (1 - \beta^{(0)})(\mathbf{P}_{t|t-1} - \mathbf{W}_t \mathbf{S}_t \mathbf{W}_t^\intercal) + \tilde{\mathbf{P}}_t, \tag{1.53}$$

where

$$\tilde{\mathbf{P}}_t = \mathbf{W}_t \left[\sum_{\theta_t > 0} \beta^{(\theta_t)} \nu_t^{(\theta_t)} (\nu_t^{(\theta_t)})^\intercal - \nu_t \nu_t^\intercal \right] \mathbf{W}_t^\intercal. \tag{1.54}$$

Let us look closer at Equation (1.53). It is the sum of the three terms. The first term reflects the possibility that no measurement is produced by the tracked target and covariance remains untouched. This happens with probability $\beta^{(0)}$. The second term represents the event, when some measurement is assumed to be produced by the target, which happens with probability $\sum_{\theta_t > 0} \beta^{(\theta_t)} = 1 - \beta^{(0)}$. The last term $\tilde{\mathbf{P}}_t$ is the so-called *spread of innovations* [13].

After the mixture reduction, the PDA filter approximates the posterior as

$$p(\mathbf{x}_t | \mathbf{Z}_{1:t}) \approx \mathcal{N}(\mathbf{x}_t, \hat{\mathbf{x}}_t, \hat{\mathbf{P}}_t). \tag{1.55}$$

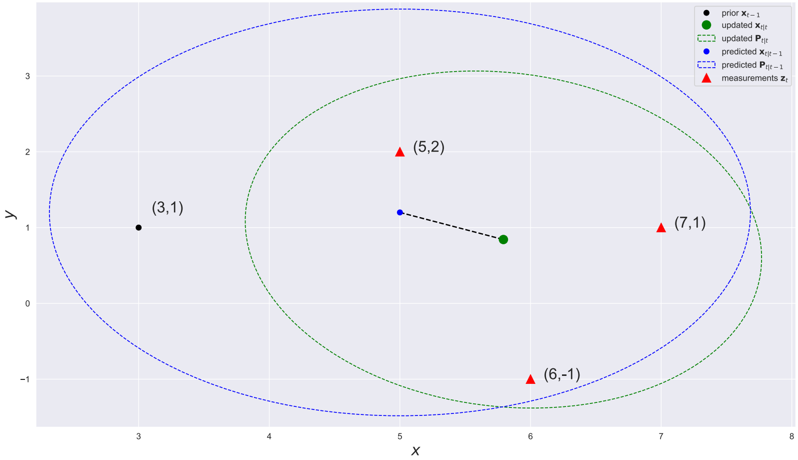


Figure 1.9: Updated state.

Returning to our example scenario depicted in the figures above, we show the updated state in Figure 1.9.

This concludes our development of the PDA filter algorithm. Let us present the pseudocode of the PDA filter. The predict step is similar to the Kalman filter prediction presented in Algorithm 1. The update step algorithm reads

Algorithm 3 PDA_Update($\mathbf{x}_{t|t-1}, \mathbf{P}_{t|t-1}, \mathbf{z}_{t|t-1}, \mathbf{S}_t, \mathbf{Z}_t, \gamma$)

```
    $\mathcal{V} \leftarrow \{\mathbf{z} \in \mathbf{Z}_t | (\mathbf{z} - \mathbf{z}_{t|t-1})\mathbf{S}_t^{-1}(\mathbf{z} - \mathbf{z}_{t|t-1}) < \gamma\}$    ▷ Elipsoidal gating
    $\beta_t^{(0)} \leftarrow \Lambda(1 - P_D)$
    $\beta_t \leftarrow \{P_D l_{i,t}(\mathbf{z}_i) | \forall \mathbf{z}_i \in \mathcal{V}\}$    ▷ Association probabilities
    $c \leftarrow \beta_t^{(0)} + \sum_i \beta_{i,t}$    ▷ Normalization constant
    $\beta_t^{(0)} \leftarrow \frac{\beta_t^{(0)}}{c}$    ▷ Normalization
    $\beta_t \leftarrow \{\frac{\beta_{i,t}}{c} | \forall \beta_{i,t} \in \beta_t\}$    ▷ Normalization
    $\nu_t \leftarrow \{\mathbf{z}_{i,t} - \mathbf{z}_{t|t-1} | \forall \mathbf{z}_{i,t} \in \mathcal{V}\}$    ▷ The innovation
    $\nu_{comb} \leftarrow \sum_i \beta_{i,t}\nu_{i,t}$    ▷ Combined innovation
    $\mathbf{W}_t \leftarrow \mathbf{P}_{t|t-1}\mathbf{H}^\intercal\mathbf{S}_t^{-1}$    ▷ The Kalman gain
    $\mathbf{x}_{t|t} \leftarrow \mathbf{x}_{t|t-1} + \mathbf{W}_t\nu_{comb}$    ▷ The updated state estimate
    $\mathbf{P}_{t|t}^C \leftarrow \mathbf{P}_{t|t-1} - \mathbf{W}_t\mathbf{S}_t\mathbf{W}_t^\intercal$
    $\tilde{\mathbf{P}}_{t|t} \leftarrow \mathbf{W}_t \left[\sum_i \beta_{i,t}\nu_t\nu_t^\intercal - \nu_{comb}\nu_{comb}^\intercal\right] \mathbf{W}_t^\intercal$    ▷ Spread of innovation
    $\mathbf{P}_{t|t} \leftarrow \beta_t^{(0)}\mathbf{P}_{t|t-1} + (1 - \beta_t^{(0)})\mathbf{P}_{t|t}^C + \tilde{\mathbf{P}}_{t|t}$    ▷ The updated covariance
    return $\mathbf{x}_{t|t},\ \mathbf{P}_{t|t}$
```

This chapter has laid the groundwork for understanding the complexities and challenges of single-target tracking (STT) in cluttered environments. We began by establishing the basic concepts of STT, including state estimation, motion models, and the impact of measurement noise. We then delved into the prevalent challenges faced by STT systems, namely clutter, missed detections, and data association ambiguities.

The chapter proceeded to introduce the theoretical framework of Bayesian filtering, demonstrating its role as the foundation for various tracking algorithms. We subsequently explored the Kalman filter, renowned for its optimality in linear Gaussian systems, and highlighted its limitations in handling clutter. To address this crucial aspect, we introduced the Probabilistic Data Association (PDA) filter, which effectively incorporates data association probabilities to mitigate the adverse effects of clutter and missed detections.

By establishing a comprehensive understanding of these fundamental concepts and algorithms, we have paved the way for exploring more advanced tracking techniques in subsequent chapters. The insights gained in this chapter will serve as a springboard for tackling the complexities of multi-target tracking and maneuvering target scenarios.

Chapter **2**

Multi-target tracking

While single-target tracking lays the foundation for understanding state estimation and filtering, real-world scenarios often involve multiple objects of interest moving simultaneously within the surveillance region. This introduces a new layer of complexity, demanding algorithms capable of handling multiple tracks, maintaining their identities, and accurately associating measurements with the corresponding targets. This chapter delves into the domain of multi-target tracking (MTT), focusing on the challenges and solutions associated with tracking multiple objects.

The necessity of MTT methods arises in scenarios where traditional single-target tracking filters prove to be insufficient. While single-target filters might be adequate when targets are well-separated and measurements can be unambiguously assigned to a specific track, such filters can be run in parallel for each target. However, when targets move in close proximity, the potential for measurement confusion increases dramatically. In these situations, we must consider the complexities of joint data association hypotheses, where a single measurement could potentially originate from multiple targets. This necessitates the use of specialized Multi-Target Tracking (MTT) methods to maintain accurate tracking and avoid misidentification. It is a common practice to employ several PDA filters when targets are distinctly separated within the surveillance area but to transition to JPDA once their validation areas begin to intersect.

Specifically, we will explore the Joint Probabilistic Data Association (JPDA) filter, which is a generalization of the PDA filter designed to address data association uncertainties in MTT scenarios. The theoretical underpinnings of the JPDA filter will be presented, followed by a detailed derivation of the practical algorithm. A critical phenomenon known as track coalescence, where closely spaced tracks tend to merge due to data association errors, will be discussed in depth. The understanding of track coalescence developed in this chapter will serve as the basis for exploring mitigation strategies in the subsequent chapter.

2.1 Joint probabilistic data association filter

There are two main assumptions that are almost always made in MTT [18]

1. **Point target assumption** Each target produces no more than a single measurement.
2. **Point measurement assumption** Every measurement is derived from no more than one target.

The implication of the point target assumption is that when dealing with a single target, our focus should be on the likelihood of individual measurements originating from that specific target, rather than considering the chance of a combination of measurements stemming from the same target. This significantly reduces computational complexity. Furthermore, without this assumption, a tracking method might be inclined to assign all measurements in a sensor scan to one target, based on previous estimates and uncertainties related to clutter intensity, target range, etc. This occurrence, known as hospitality to clutter [20], is much less probable when the point target assumption is applied. Concerning the assumption of point measurement, it also diminishes the computational complexity of the problem. In the event that a single measurement could be generated by several targets, we could encounter a scenario in which we need to consider that all the targets being tracked could have generated this specific measurement. This situation exponentially increases complexity with increasing number of targets and intensity of clutter. Both of these assumptions are dubious from a physical perspective [18], however, we assume that the sensor used has some specific features that force the output to satisfy these assumptions.

These two assumptions are fundamental for multi-target tracking. However, there is a so-called *standard model of multi-target tracking* which contains the following assumptions

1. Markov assumption for state transition.
2. New targets are born according to a Poisson process with intensity $\mu(\mathbf{x})$.
3. Existing targets survive from time step $t-1$ to t with probability $P_s(\mathbf{x}_{t-1})$.

4. The motion of a surviving target is given by $\mathbf{f}_{\mathbf{x}}(\mathbf{x}_t|\mathbf{x}_{t-1})$.
5. A target with state $\mathbf{x}_t$ generates a measurement $\mathbf{z}_t$ with probability $P_D(\mathbf{x}_t)$.
6. Clutter measurements occur according to a Poisson process with intensity $\lambda(\mathbf{z})$.
7. The measurement of a detected target is related to the state according to $\mathbf{f}_{\mathbf{z}}(\mathbf{z}_t|\mathbf{x}_t)$.

The implications of Assumptions 2. and 3. are that over time, multiple targets will be detected in the surveillance area. This also introduces ambiguity about the exact number of actual targets within the monitored region. Various strategies exist to manage this ambiguity, such as the IPDA filter [21] or the well-known M/N rule [17]. However, addressing this particular type of ambiguity is beyond the scope of this thesis, and we will not consider this uncertainty in further derivations.

The JPDA is a special case of the standard MTT model, where

8. The count of targets, n, remains steady and is known (i.e., zero birth intensity and death probability).
9. The single-target motion model and the likelihood are Gaussian and linear:

$$\mathbf{f}_{\mathbf{x}}(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \mathbf{F}\mathbf{x}_{t-1}, \mathbf{Q}) \tag{2.1}$$

$$\mathbf{f}_{\mathbf{z}}(\mathbf{z}_k|\mathbf{x}_k) = \mathcal{N}(\mathbf{z}_k; \mathbf{H}\mathbf{x}_t, \mathbf{R}) \tag{2.2}$$

10. The prior densities of the targets are independent and Gaussian.

$$p^i_{t-1}(\mathbf{x}^i_{t-1}) = \mathcal{N}(\mathbf{x}^i_{t-1}; \hat{\mathbf{x}}^i_{t-1}, \mathbf{P}^i_{t-1}). \tag{2.3}$$

11. The target i has constant detection probability P^i_D.
12. The clutter Poisson process has a constant intensity λ.

Assumption 10 reveals the parallelism between JPDA and PDA. The objective of JPDA is to merge all association hypotheses into a single Gaussian for each target. However, this is considerably more challenging for JPDA, as the number of association hypotheses grows exponentially compared to PDA. Figure 2.1 shows the association hypotheses for a two target case.

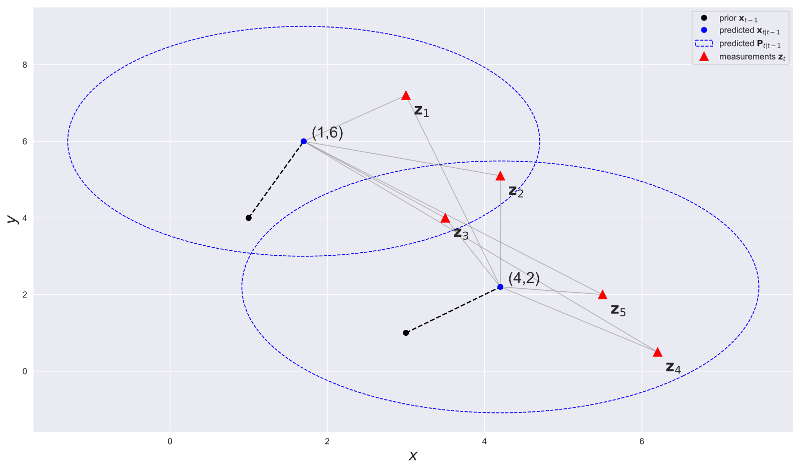


Figure 2.1: Association hypotheses for two targets and 5 measurements..

JPDA has the same algorithmic cycle: prediction, data association, and update. Let us go through these stages and outline the main differences.

The prediction step is pretty similar and based on the Chapman-Kolmogorov equation (1.12) as in (1.28)

$$p^i_{t|t-1}(\mathbf{x}^i_t) = \int \mathbf{f}_{\mathbf{x}}(\mathbf{x}^i_t|\mathbf{x}^i_{t-1})p_t(\mathbf{x}^i_{t-1})d\mathbf{x}^i_{t-1}. \tag{2.4}$$

$$= \mathcal{N}(\mathbf{x}^i_t; \hat{\mathbf{x}}^i_{t|t-1}, \mathbf{P}^i_{t|t-1}), \tag{2.5}$$

where $\hat{\mathbf{x}}^i_{t|t-1}$ and $\mathbf{P}^i_{t|t-1}$ are defined in Equation (1.37) and Equation (1.38) respectively. For measurement prediction $\hat{\mathbf{z}}^i_{t|t-1}$ and its covariance $\mathbf{S}^i_{t|t-1}$, we use Equations (1.39) and (1.40) respectively.

In PDA we had the variable θ_t defined in (1.26). For the joint case we must define such variable for each target, so we define the vector $\Theta_t = [\theta^1_t, \theta^2_t, \ldots, \theta^n_t]$ such that

$$\theta^i_t = \begin{cases} j & \text{if measurement } j \text{ is produced by the target } i \\ 0 & \text{if no measurement is produced by the target } i. \end{cases} \tag{2.6}$$

Note that one realization of vector Θ_t represents one joint hypothesis, where each target is assigned with either a real or dummy measurement.

Again, similarly as in Equation (1.27), we use the total probability theorem to express the posterior. Note that the association hypotheses are mutually exclusive and exhaustive [18] thanks to the point assumptions mentioned at the beginning of this chapter. This yields

$$p(\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n | \mathbf{Z}_{1:t}) = \sum_{\Theta_t} p(\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n | \Theta_t, \mathbf{Z}_{1:t}) Pr\{\Theta_t | \mathbf{Z}_{1:t}\}. \tag{2.7}$$

The sum goes over all feasible association hypotheses. Note that the cardinality of Θ_t grows exponentially with the increase of n and m_t. In general, ellipsoidal gating is always performed in JPDA. The first obvious reason is computational optimization. Furthermore, computing densities does not make sense for measurements far from prediction, since the corresponding probabilities $Pr\{\Theta_t|\mathbf{Z}_{1:t}\}$ would quickly approach zero. We would not reflect ellipsoidal gating in the subsequent derivations for the sake of simplicity.

In order to find the posterior (2.7), we need to find the expression for event-conditional densities $p(\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n | \Theta_t, \mathbf{Z}_{1:t})$ and association posterior probabilities $Pr\{\Theta_t|\mathbf{Z}_{1:t}\}$. For the initial term, the process is relatively straightforward - if the measurement $\mathbf{z}_t^{\theta_t^i}$ is assigned to the target i, we employ the conventional KF formula, otherwise, we use the prediction. The equation is as follows [18]

$$p(\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n | \Theta_t, \mathbf{Z}_{1:t}) \propto \prod_{i:\theta_t^i = 0} p_{t|t-1}^i(\mathbf{x}_t^i) \prod_{i:\theta_t^i > 0} \mathbf{f}_{\mathbf{z}}(\mathbf{z}_t^{\theta_t^i} | \mathbf{x}_t^i) p_{t|t-1}^i(\mathbf{x}_t^i). \tag{2.8}$$

In order to define the hypothesis probability $Pr\{\Theta_t|\mathbf{Z}_{1:t}\}$ we need to define the track-to-measurement likelihood [18]

$$l^{i,\theta_t^i} = \int \mathbf{f}_{\mathbf{z}}(\mathbf{z}_t^{\theta_t^i} | \mathbf{x}_t^i) p_{t|t-1}^i(\mathbf{x}_t^i) d\mathbf{x}_t^i = \mathcal{N}(\mathbf{z}_t^{\theta_t^i}; \hat{\mathbf{z}}_{t|t-1}^i, \mathbf{S}_{t|t-1}^i). \tag{2.9}$$

Then the probabilities for the association hypotheses are given by [18]

$$Pr\{\Theta_t | \mathbf{Z}_{1:t}\} \propto \prod_{i:\theta_t^i = 0} \lambda(1 - P_D) \prod_{i:\theta_t^i > 0} P_D l^{i,\theta_t^i}. \tag{2.10}$$

Returning to the example with two targets from Figure 2.1, we present calculation of individual association probabilities in Table 2.1.

However, the key principle of both the PDA and the JPDA filters is to merge different hypotheses for each track. Now, we need to marginalize the association probabilities $Pr\{\Theta_t|\mathbf{Z}_{1:t}\}$ as follows

$$\beta_t^{i,j} = Pr\{\mathbf{z}_t^j \text{ is assigned to target } i | \mathbf{Z}_{1:t}\} = \sum_{\Theta_t : \theta_t^i = j} Pr\{\Theta_t | \mathbf{Z}_{1:t}\}. \tag{2.11}$$

This means that we sum up the probabilities for all feasible hypotheses in which the current measurement is assigned to the specific target. In our

Θ	$Pr\{\Theta_t \vert \mathbf{Z}_{1:t}\}$
$[0,0]$	$\lambda^2(1-P_D)^2$
$[0,1]$	$\lambda(1-P_D)\cdot P_D l^{2,1}$
$[0,2]$	$\lambda(1-P_D)\cdot P_D l^{2,2}$
$[0,3]$	$\lambda(1-P_D)\cdot P_D l^{2,3}$
$[0,4]$	Out of gate for the second target
$[0,5]$	Out of gate for the second target
...	...
$[2,3]$	$P_D^2 \cdot l^{1,2}\cdot l^{2,3}$
...	...

Table 2.1: Calculation of association posterior probabilities.

example,

$$\begin{aligned}\beta^{1,1} = Pr\{[1,0]|\mathbf{Z}_{1:t}\} + Pr\{[1,1]|\mathbf{Z}_{1:t}\} + Pr\{[1,2]|\mathbf{Z}_{1:t}\}+\\ + Pr\{[1,3]|\mathbf{Z}_{1:t}\} + Pr\{[1,4]|\mathbf{Z}_{1:t}\}+\\ + Pr\{[1,5]|\mathbf{Z}_{1:t}\}.\end{aligned} \tag{2.12}$$

Now, the only thing left is to describe the update step of JPDA. In fact, the JPDA update step is the PDA update performed for all n targets using the association probabilities calculated above. We define Kalman gain for i-th track as follows

$$\mathbf{W}_t^i = \mathbf{P}_{t|t-1}^i \mathbf{H}^\intercal (\mathbf{H}\mathbf{P}_{t|t-1}^i \mathbf{H}^\intercal + \mathbf{R})^{-1} = \mathbf{P}_{t|t-1}^i \mathbf{H}^\intercal (\mathbf{S}_t^i)^{-1}. \tag{2.13}$$

Then we express the posterior event-conditional expectations and covariances as

$$\hat{\mathbf{x}}_t^{i,\theta_t^i} = \begin{cases} \hat{\mathbf{x}}_{t|t-1}^i & \text{if } \theta_t^i = 0 \\ \hat{\mathbf{x}}_{t|t-1}^i + \mathbf{W}_t^i(\mathbf{z}_t^{\theta_t^i} - \hat{\mathbf{z}}_{t|t-1}^i) & \text{if } \theta_t^i > 0 \end{cases} \tag{2.14}$$

$$\mathbf{P}_t^{i,\theta_t^i} = \begin{cases} \mathbf{P}_{t|t-1}^i & \text{if } \theta_t^i = 0 \\ (\mathbf{I} - \mathbf{W}_t^i\mathbf{H})\mathbf{P}_{t|t-1}^i & \text{if } \theta_t^i > 0. \end{cases} \tag{2.15}$$

Using these expressions, we can rewrite (2.8) as

$$p(\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n|\Theta_t, \mathbf{Z}_{1:t}) = \prod_{i=1}^{n} \mathcal{N}(\mathbf{x}_t^i; \hat{\mathbf{x}}_t^{i,\theta_t^i}, \mathbf{P}_t^{i,\theta_t^i}). \tag{2.16}$$

Finally, using the posterior association probabilities calculated from Equation (2.11), we can find the single Gaussian posterior approximation for each track according to

$$p_t^i(\mathbf{x}_t^i) = \sum_{j=0}^{m_t} \beta_t^{i,j} \mathcal{N}(\mathbf{x}_t^i; \hat{\mathbf{x}}_t^{i,j}, \mathbf{P}_t^{i,j}) \approx \mathcal{N}(\mathbf{x}_t^i; \hat{\mathbf{x}}_t^i, \mathbf{P}_t^i). \tag{2.17}$$

That is, the posterior of each track is a Gaussian mixture, which is reduced in exactly the same manner as in the PDA filter. To find the merged expectation and covariance, we use equations (1.47) and (1.53), respectively.

Essentially, the main difference between PDA and JPDA is the calculation of event-conditional densities and association probabilities. Let us also present the practical implementation of this calculation. This approach was used for JPDA implementation for experiments described in Chapter 4. We will only describe the computation of association probabilities $\beta_t^{i,j}$ from Equation (2.11).

First, we need to construct the cost matrix $\mathbf{C} \in \mathbb{R}^{n,m_t+n}$. This matrix is separated into two blocks column-wise. The first block comprises of track-to-measurement likelihoods $P_D l^{i,j}$. The second block is a diagonal matrix with the expression $\lambda(1-P_D)$ representing track misdetections. Hence, matrix has the following form

$$\mathbf{C}_t = \left[\begin{array}{ccc|ccc} P_D l^{1,1} & \dots & P_D l^{1,m_t} & \lambda(1-P_D) & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ P_D l^{n,1} & \dots & P_D l^{n,m_t} & 0 & \dots & \lambda(1-P_D) \end{array}\right] \tag{2.18}$$

Then, for each feasible hypothesis $\Theta_{i,t}$, we construct the assignment matrix $\mathbf{A}_t^{\Theta_{i,t}} \in \{0,1\}^{n,m_t+n}$. This matrix has exactly one 1 in each row. 1 indicates a target assignment. If the measurement j was assigned to the target k, then $\mathbf{A}_t^{\Theta_{i,t}}[k,j] = 1$. In case of misdetection $\mathbf{A}_t^{\Theta_{i,t}}[k,n+k] = 1$. Every other number in this matrix is 0. Then we can calculate the association probability from equation (2.10) by

$$Pr\{\Theta_{i,t}|\mathbf{Z}_t\} = \mathcal{P}((\mathbf{A}_t^{\Theta_{i,t}})^\intercal \mathbf{C}_t) \tag{2.19}$$

Where $\mathcal{P}(\cdot)$ stands for product operator and means the product of positive elements on the main diagonal, that is $\mathcal{P}(A) = \sum_{i:a_{ii}>0}^{n} a_{ii}$. To obtain the marginal probabilities $\beta_t^{i,j}$, we sum the probabilities of all hypotheses where the measurement j is assigned to the target i according to Equation (2.11). For our example, the calculation of $Pr\{[1,5]|\mathbf{Z}_{1:t}\}$ proceeds as follows

$$\mathbf{C}_t = \left[\begin{array}{ccc|cc} P_D l^{1,1} & \dots & P_D l^{1,5} & \lambda(1-P_D) & 0 \\ P_D l^{2,1} & \dots & P_D l^{2,5} & 0 & \lambda(1-P_D) \end{array}\right], \tag{2.20}$$

$$\mathbf{A}_t^{[1,5]} = \left[\begin{array}{ccccc|cc} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right], \tag{2.21}$$

$$\begin{aligned} Pr\{[1,5]|\mathbf{Z}_{1:t}\} &= \mathcal{P}((\mathbf{A}_t^{[1,5]})^\intercal \mathbf{C}_t) \\ &= \mathcal{P}\left(\left[\begin{array}{ccc|cc} P_D l^{1,1} & \dots & P_D l^{1,5} & \lambda(1-P_D) & 0 \\ 0 & \dots & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 \\ P_D l^{2,1} & \dots & P_D l^{2,5} & 0 & \lambda(1-P_D) \\ 0 & \dots & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 \end{array}\right]\right) \\ &= P_D l^{1,1} \cdot P_D l^{2,5}. \end{aligned} \tag{2.22}$$

The JPDA algorithm provides an advanced solution for managing uncertainties in data association in scenarios involving the tracking of multiple targets. Nevertheless, a specific difficulty emerges when targets are in close proximity – the coalescence issue. As the tracks get spatially closer, conventional JPDA may find it difficult to preserve their unique identities, resulting in possible track fusion or misassociations.

2.2 Coalescence problem

Track coalescence occurs when multiple targets move in close proximity with overlapping validation gates. In such scenarios, the probabilistic averaging approach of JPDA assigns similar association probabilities to multiple track-measurement pairings. Consequently, the filter updates the state estimates for both (or more) targets towards a point between their actual positions. This merging of tracks can lead to inaccurate state estimation and potential track loss, particularly when the targets diverge after a period of close proximity.

This effect is a direct consequence of the manner in which the JPDA filter performs soft data association [22]. Mathematically, when the targets are close to each other, the predicted marginal posterior distributions $p_t^i(\mathbf{x}_t^i)$ for each target i become similar. This similarity arises because the targets' states, represented by their positions and possibly other motion-related parameters, are nearly identical in the state space, leading to similar posterior distributions based on the prior information.

The JPDA filter calculates the approximate marginal posterior distribution by summing over all possible associations $\Theta_t^{(i)}$ for each target i, as shown in the equation (2.7). When the predicted marginal posteriors are similar, the summation in the above equation results in approximate marginal posteriors that are also similar for closely spaced targets. Consequently, the Minimum Mean Square Error (MMSE) state estimates derived from these approximate marginal posteriors tend to be similar for all targets, leading to the coalescence of the estimated tracks. This effect is particularly pronounced when the states of the targets are so close that the uncertainty of measurement origin causes the association probabilities to be nearly uniform across the targets, further exacerbating the coalescence effect [22].

Several factors exacerbate the track coalescence problem

- **High Clutter Density** λ The presence of clutter increases the uncertainty in data association, further contributing to erroneous track updates and a higher likelihood of coalescence.

- **Low Detection Probability** P_D Misdetections lead to larger validation gates, increasing the chances of overlapping gates and subsequent coalescence.

- **Similar Target Motion** Targets moving with similar velocities and trajectories are more prone to track coalescence because of the difficulty of distinguishing their individual contributions to the measurements.

The consequences of track coalescence can be detrimental to tracking performance.

- **Loss of Track Identity** The merging of tracks can lead to the loss of individual target identities, making it difficult to distinguish and track targets after they separate.

- **Degraded State Estimation** The averaged state estimates resulting from track coalescence deviate from the true target positions, affecting the accuracy of subsequent predictions and filtering updates.

- **Increased Track Loss Probability** As the state estimates become less accurate, the likelihood of track loss increases, especially when the targets maneuver or change direction after coalescence.

Although the theoretical consequences of track coalescence are clear, examining real-world scenarios provides a deeper understanding of its impact on tracking performance and the situations where it poses a critical challenge versus instances where it might be less consequential.

In air traffic control (ATC) systems, maintaining accurate tracks and unique identities for each aircraft is crucial to ensure safe separation and preventing collisions. Track coalescence, even for a brief period, can have catastrophic consequences, such as

- **Loss of Separation** Merged tracks can mask the true positions and trajectories of aircraft, potentially leading to a loss of required separation and an increased risk of mid-air collisions.
- **Misidentification** Inaccurate state estimates resulting from coalescence can cause misidentification of aircraft, leading to incorrect instructions and endangering flight safety.
- **Confusion and Delays** Resolving coalesced tracks requires additional time and effort, causing confusion for air traffic controllers and potentially leading to delays and disruptions in flight operations.

Therefore, in ATC systems, minimizing track coalescence is paramount for maintaining the safety and efficiency of air traffic operations.

In contrast, consider wildlife monitoring applications where tracking a population of animals is the primary goal. While maintaining distinct tracks for each individual might be ideal, temporary track coalescence might not be a significant issue. In many wildlife studies, understanding the overall population dynamics and movement patterns is more important than tracking individual animals. Coalescence may not significantly impact this analysis, especially if the animals eventually separate and their tracks are re-established.

Furthermore, even with coalesced tracks, researchers can often rely on additional information, such as size, color, or movement patterns, to differentiate species and maintain relevant ecological insights. And finally, wildlife monitoring often involves limited resources and observation capabilities. Accepting some degree of track coalescence might be necessary to balance tracking accuracy with feasibility and cost-effectiveness.

Therefore, while track coalescence remains undesirable, it might not pose a critical challenge in applications where the focus is on broader population-level understanding rather than individual identification and precise tracking.

These real-world examples highlight the varying impact of track coalescence depending on the specific application and its requirements. The next chapter will explore strategies to address this challenge and introduce our novel method to achieve robust and efficient track maintenance in the presence of coalescence.

Chapter 3

Coalescence suppression

As established in previous chapters, track coalescence presents a significant challenge for multi-target tracking systems, particularly those utilizing JPDA filters. This chapter delves into the various techniques developed to combat this phenomenon, reviewing their strengths and limitations. We will then present a novel approach that combines adaptive validation gating with strategic hypothesis pruning, offering a robust and computationally efficient solution for maintaining track integrity in the presence of closely spaced targets.

3.1 Exact nearest neighbor

One of the two main paradigms in tackling the coalescence problem is hypothesis pruning. The exact nearest-neighbor PDA (ENNPDA) presented in [15] is a very simple yet powerful approach. Originally, it was designed to run in real time on a microprocessor, so the original PDA was optimized, and

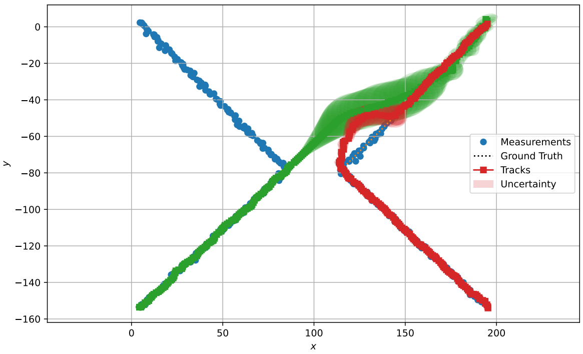


Figure 3.1: Track coalescence example.

pruning was introduced both to address the coalescence problem and to lower the computational cost.

Basically, once the probabilities $\beta_t^{i,j}$ from Equation (2.11) are calculated, ENNPDA employs a nearest neighbor approach. This means that each measurement is assigned to the track with the highest association probability, creating a one-to-one correspondence. Unlike standard nearest-neighbor methods that solely rely on the distance between a track's predicted position and a measurement, ENNPDA factors in the association probability. This leads to more accurate associations, especially in situations where tracks and measurements are close to each other.

Let us take a closer look at the set of hypotheses Θ_t. We will use the notation of [16]. We first define a target detection vector $\phi_t \in \{0,1\}^n$ as follows

$$\phi_{j,t}(\Theta_t^i) = \begin{cases} 1 & \text{if the } j\text{th target was detected in hypothesis } \Theta_t^i, \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

Next, we define the measurement association vector $\psi_t \in \{0,1\}^{m_t}$ as

$$\psi_{j,t}(\Theta_t^i) = \begin{cases} 1 & \text{if the } j\text{-th measurement belongs to detected target} \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

We also define the operator $D(x) = \sum_i x_i$. Then $D(\phi_t(\Theta_t^i))$ stands for the number of detected targets, and $D(\psi_t(\Theta_t^i))$ stands for the number of measurements associated with the detected tracks.

Then we define the sequence of permutation matrices $\{\chi_t\}$, which randomly permutes the measurements to match them with the detected targets. We will not delve into the mathematical details of this permutation.

Using this notation in [16] we split the series of association hypotheses $\{\Theta_t\}$ into two subsets: (ϕ, ψ)-hypotheses and (ϕ, χ)-hypotheses. (ϕ, ψ)-hypothesis is about the classification of measurements as target-originated or clutter, while the (ϕ, χ)-hypothesis is about the ordering of measurements to targets.

In ENNPDA, we take all feasible association hypotheses $\{\Theta_t\}$ that satisfy $D(\psi_t(\Theta_t^i)) = D(\phi_t(\Theta_t^i)) \leq \min\{n, m_t\}$ and select one that

$$\hat{\Theta}_t = \arg\max_{\Theta_t} = Pr\{\Theta_t | \mathbf{Z}_{1:t}\}. \tag{3.3}$$

That is, we prune all (ϕ, χ)-hypotheses except for one $\hat{\Theta}_t$ with the highest likelihood and use it for update. The resulting ENNPDA appeared to be remarkably insensitive to track coalescence in case there is no clutter and no missed detections. However, the dramatic pruning used for ENNPDA leads to an undesirable sensitivity to clutter and missed detections [23]. As a result, by using the ENNPDA coalescence suppression method, we lose the robust tracking performance of JPDA, which is undesirable. Nevertheless, the ENNPDA approach pioneered the pruning strategy to mitigate coalescence and laid the foundation for the development of more advanced pruning techniques.

3.2 JPDA*

In order to benefit from pruning while maintaining the insensitivity of JPDA to clutter and missed detections, the JPDA* approach was introduced [16]. It was found that track coalescence could be avoided by pruning χ-hypotheses. In particular, we prune the permutations of the hypothesis with the highest likelihood in order to reduce the uncertainty for resulting posterior.

In order to keep the filter robust to clutter and misdetections, we must not prune any ϕ or ψ hypotheses [16]. The combination of these two findings applied to JPDA is called JPDA*.

In JPDA*, we first evaluate all feasible (ϕ_t, ψ_t)-hypotheses, then per (ϕ_t, ψ_t)-hypothesis we prune all less likely χ-hypotheses, that is, permutations of the hypothesis with the highest likelihood. Mathematically, for every ϕ_t and ψ_t, satisfying $D(\phi_t) = D(\psi_t) \leq \min\{n, m_t\}$ we look for

$$\hat{\chi}_t = \arg\max_{\chi} \ Pr\{\Theta_t^{(\phi_t,\psi_t)} | \mathbf{Z}_{1:t}\}. \tag{3.4}$$

Then we prune all χ-hypotheses for given ϕ_t and ψ_t except for the $\hat{\chi}_t$-hypothesis. Obviously, the resulting probabilities must be normalized after pruning.

Simulations performed in [16] show that JPDA* outperforms PDA, JPDA, and ENNPDA in suppressing coalescence, while remaining robust to misdetections and clutter. To demonstrate the difference between these methods, Table 3.1 displays the manner in which each approach prunes the hypotheses in a particular scenario.

3.3 Coupling

One of the approaches to tackle the colaescence is to make use of the concept of a joint state for multiple targets. For example, JPDAC [17]. The effectiveness of the JPDAC approach in combination with two other Bayesian approaches (imaging sensor filter and interactive multiple model - IMM, respectively) has been demonstrated for closely spaced target situations in [24], [25].

Θ	$Pr\{\Theta\|\mathbf{Z}_{1:t}\}$	ENNPDA	JPDA*
$[0,0]$	0.00011	×	
$[0,1]$	0.05758	×	× in favor of $[1,0]$
$[0,2]$	0.31308		
$[1,0]$	0.11829	×	
$[1,2]$	0.17577	×	× in favor of $[2,1]$
$[2,0]$	0.13885	×	× in favor of $[0,2]$
$[2,1]$	0.19632	×	

Table 3.1: Comparison of pruning of ENNPDA and JPDA* for two targets and two received measurements. Symbol × represent that the corresponding hypothesis was pruned by the algorithm.

Another method that uses the idea of joint state is coupled PDA (CPDA) introduced in [16]. Furthermore, the authors combined this approach with hypotheses pruning resulting in CPDA* [16]. From their results, it is evident that hypothesis pruning appears to be the most efficient method to suppress the coalescence effect (CPDA* outperforms CPDA, CPDA* and JPDA* show similar performance).

Evidence has been presented that ENNPDA, JPDA*, and CPDA* avoid track coalescence by positioning a Gaussian density around a local optimum (not necessarily the global optimum). This method seems so efficient that the requirement to recall the coupling between tracks practically disappears [16].

3.4 Bias removal

Another approach to coalescence suppression is based on the removal of bias from the posterior state estimate. The state estimate becomes biased when the targets approach each other and their validation regions intersect, since any track attempting to follow a particular target will be attracted, to some extent, towards a neighboring target with whose returns it correlates [26]. Furthermore, the magnitude of the tracking bias is quantitatively analyzed in [26]. The paper establishes that the bias increases as the distance between the targets decreases, reaching a maximum when the targets are very close to each other. Using a two-target scenario, the authors show that target separations smaller than a certain critical value tend to cause a total coalescence of the tracks.

In the latter works, this paradigm was used for Bias Removal JPDA (BR-JPDA) [27]. The authors analyze the causes of state bias in JPDA and provide a direct computation equation for the bias in an ideal two-target case. They then extend the bias estimation to more general and practical cases by defining target detection hypotheses and target-to-target association hypotheses. The estimated bias is subsequently removed from the state updated by JPDA to produce an unbiased state.

However, as discovered by [26], the root of bias lies in the measurements generated by adjacent tracks, which are used to update the current track. Therefore, to estimate the bias, it is essential to identify the sources of the given measurements and exclude those originating from neighboring tracks from our considerations, which is the main problem. If not, the bias estimation relies on an approximation that inherently holds uncertainty. However, the authors show that BRJPDA [27] can effectively handle track coalescence and offers performance similar to that of JPDA* and outperforms that of ”vanilla” JPDA.

We examined prevalent methods for coalescence prevention in JPDA filters. Among these, hypotheses pruning, particularly JPDA*, appears to be the most straightforward and effective for this purpose. It surpasses strategies that rely on coupling, including the CPDA* that employs both coupling and hypothesis pruning. Essentially, hypothesis pruning forces the JPDA to make a decision [18]. We chose to expand on this concept by incorporating adaptive validation gating.

3.5 Adaptive validation gating

In this section we present our contribution to the field of track coalescence suppression. The approach leverages the concept of pruning introduced in JPDA* and dynamically adjusts the size of the validation gate to mitigate the risk of track coalescence.

The proposed method introduces an adaptive validation gating technique that scales the gating probability, P_G, based on the spatial proximity of the track validation ellipses. We introduced the gating probability in Section 1.5. The default value of P_G (which is typically equal to one or nearly so) is dynamically reduced toward some value denoted $P_{squeeze}$ when the validation ellipses of multiple tracks intersect, effectively shrinking the validation gate. This reduction in gate size aims to limit the association possibilities between closely spaced tracks, thus reducing the likelihood of coalescence. It is important to note that we only apply gate squeezing to pairs of gates that intersect in the current time scan t.

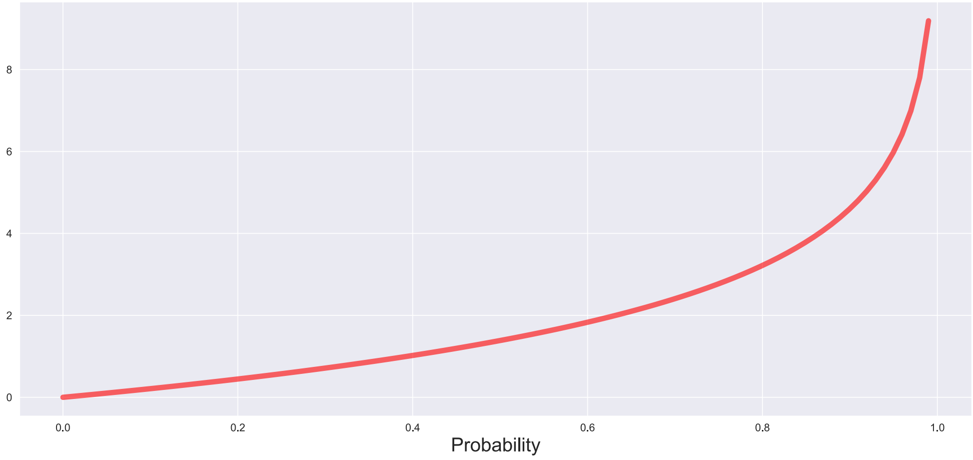


Figure 3.2: χ^2 with two degrees of freedom percentile function.

So, the gating process from (1.41) is modified as follows

$$\mathcal{V}(\mathbf{Z};\gamma(P_{G,t})) = \{\mathbf{z} \in \mathbf{Z} | (\mathbf{z} - \hat{\mathbf{z}}_{t|t-1})\mathbf{S}_t^{-1}(\mathbf{z} - \hat{\mathbf{z}}_{t|t-1}) < \gamma(P_{G,t})\}. \tag{3.5}$$

$$\gamma(P_{G,t}) = \begin{cases} F^{-1}_{\chi^2_{df=l}}(P_G) & \text{if no intersection is detected} \\ F^{-1}_{\chi^2_{df=l}}(P_{squeeze}) & \text{for intersected gates,} \end{cases} \tag{3.6}$$

where $F^{-1}_{\chi^2_{df=l}}(\cdot)$ stands for inverse CDF function, or percentile function of χ^2 distribution with l degrees of freedom, where l is the length of the measurement vector. In Figure 3.2, the percentile function of the χ^2 distribution with two degrees of freedom is depicted. It is evident that even a minor adjustment in the desired probability significantly reduces the output of the function.

Reducing the probability of gating has consequences. To prevent excessive gate shrinkage and the subsequent risk of missed detections, a control mechanism is implemented. Two key parameters govern this mechanism.

- `squeeze_threshold` This parameter defines the maximum number of consecutive time steps during which gate squeezing can be applied. Once this threshold is reached, the gate size is no longer reduced.
- `squeeze_cooldown` This parameter specifies the number of time steps during which gate squeezing is prohibited after reaching the `squeeze_threshold`. This cooldown period allows tracks to diverge and reduces the risk of track loss due to overly restrictive gating.

The algorithm operates as follows.

1. **Intersection Detection** At each time step, the algorithm checks for intersections between the validation ellipses of all active tracks.
2. **Gate Size Adaptation** If intersections are detected, the value of P_{G_t} for the corresponding tracks is reduced, affecting $\gamma(P_{G,t})$, effectively shrinking the validation gates.
3. **Threshold and Cooldown Management** The algorithm tracks the number of consecutive squeezing instances. If the `squeeze_threshold` is reached, gate squeezing is halted for the duration specified by the `squeeze_cooldown` parameter.
4. **Hypothesis pruning** Further reduction of hypotheses by implementing the JPDA* pruning presented in Section 3.2.

Pseudocode is presented in Algorithm 4.

This adaptive validation gating method offers several potential benefits for JPDA filtering

- **Reduced Track Coalescence** By dynamically adjusting the validation gate size, the proposed method aims to minimize the likelihood of track coalescence, particularly in scenarios with closely spaced objects.
- **Improved Tracking Accuracy** By mitigating track coalescence, the method is expected to enhance the overall tracking accuracy and maintain the individual identities of closely spaced objects.
- **Adaptive and Responsive Behavior** The use of thresholds and cooldown periods ensures that the gate size adaptation is responsive to the dynamic nature of the tracking environment, preventing excessively restrictive gating and potential track loss.

This section presented a novel approach to track coalescence suppression in JPDA filters. The subsequent sections of the thesis will delve into a detailed analysis of the method's performance through simulations and experiments. These analyses will evaluate the effectiveness of the proposed approach in comparison to existing techniques and explore the impact of various parameter choices on tracking performance. Furthermore, the limitations of the method and potential avenues for future research will be discussed.

Algorithm 4 Adaptive Validation Gating with Hypothesis Pruning

```
Require: Tracks $T$, measurements $Z$, gating probability $P_G$, reduced gating
    probability $P_{squeeze}$, squeeze_threshold $S_T$, squeeze_cooldown $S_C$
Ensure: Updated tracks with reduced coalescence
Initialize $squeeze\_count \leftarrow 0$ for each track
for each time step $t$ do
    for each track $T_i$ do
        Generate validation gate based on $P_G$
        for each pair of tracks $T_i$, $T_j$ with intersecting gates do
            if $squeeze\_count[T_i] < S_T$ and $squeeze\_count[T_j] < S_T$ then
                $P_G[T_i] \leftarrow P_{squeeze}$
                $P_G[T_j] \leftarrow P_{squeeze}$
                $squeeze\_count[T_i] \leftarrow squeeze\_count[T_i] + 1$
                $squeeze\_count[T_j] \leftarrow squeeze\_count[T_j] + 1$
            end if
        end for
        Associate measurements within the gate to $T_i$
        Calculate association probabilities using JPDA
    end for
    Perform JPDA* hypothesis pruning
    Update tracks using JPDA
    for each track $T_i$ with $squeeze\_count[T_i] \geq S_T$ do
        if $cooldown\_timer[T_i] = 0$ then
            $squeeze\_count[T_i] \leftarrow 0$
            $cooldown\_timer[T_i] \leftarrow S_C$
        else
            $cooldown\_timer[T_i] \leftarrow cooldown\_timer[T_i] - 1$
        end if
    end for
end for
```

Chapter 4

Experiment validation

This chapter focuses on the evaluation and comparison of three JPDA-based algorithms: the standard JPDA from Section 2.1, JPDA* from Section 3.2, and a novel variant we propose, JPDA* with adaptive gating from Section 3.5. Our proposed method enhances the JPDA* filter by dynamically adjusting the validation gate size based on the proximity and estimated states of nearby targets. This adaptive gating mechanism aims to further reduce the likelihood of track coalescence while maintaining tracking accuracy.

The goal of this chapter is twofold. Firstly, our objective is to identify the optimal parameters for our method using Monte Carlo simulations. Secondly, we will conduct a comprehensive comparison of the three algorithms across different tracking scenarios to assess their performance and coalescence mitigation capabilities. Through this analysis, we aim to demonstrate the effectiveness of our proposed method and its potential for improved multi-target tracking in challenging environments. Subsequently, we will evaluate these algorithms across three distinct scenarios, each of which presents varying levels of complexity.

4.1 Methodology

Three algorithms were implemented in Python, using the NumPy and SciPy libraries. All filters use the constant velocity model presented in Section 1.1 with a 4 dimensional state $\mathbf{x} = [x,\ v_x,\ y,\ v_y]^\intercal$. The measurement model matrix $\mathbf{H}$ is compatible with this state. For the process noise covariance we

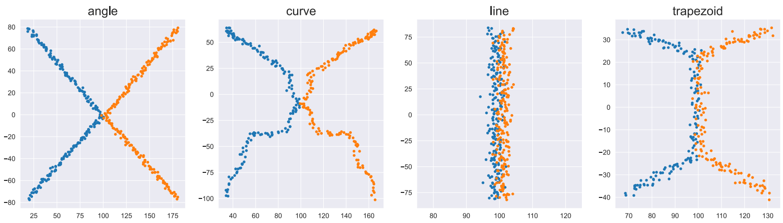


Figure 4.1: Synthetic trajectories for simulations.

use

$$\mathbf{Q} = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.1}$$

anticipating greater variability in velocities compared to positions. The implementations are available on the attached media. All simulations were executed on MacBook Pro 2020 with M1 chip with

- 8 CPU cores,
- 3.2 GHz maximum CPU clock rate,
- 8 GB RAM.

Dataset

We created four synthetic trajectories for the simulations shown in Figure 4.1. All targets move with constant velocity. The base trajectory for one target is created using blender software and then mirrored according to the parameter `mirror_offset` that defines the distance between measurements at the closest point for two tracks as shown in Figure 4.2. All trajectories are centered around $x = 100$.

A `Radar` class has been developed to emulate the data flow. In addition to setting the trajectory type and `mirror_offset`, it allows configuration of the clutter rate λ, the dimensions of the surveillance area, and the measurement covariance matrix $\mathbf{R}$ (see Figures 4.3 and 4.4).

Metrics

To evaluate and compare the performance of the JPDA, JPDA*, and our method with adaptive gating algorithms, we employ two primary metrics:

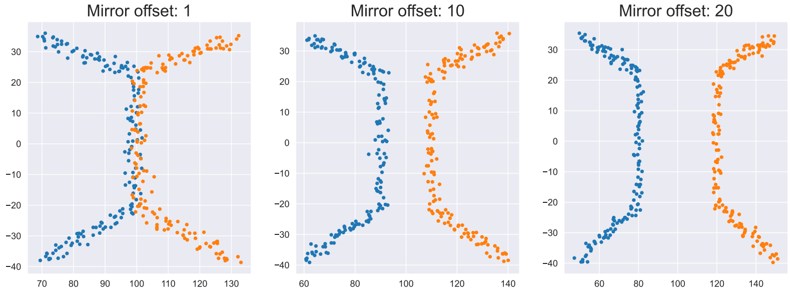


Figure 4.2: Impact of mirror offset on trajectory.

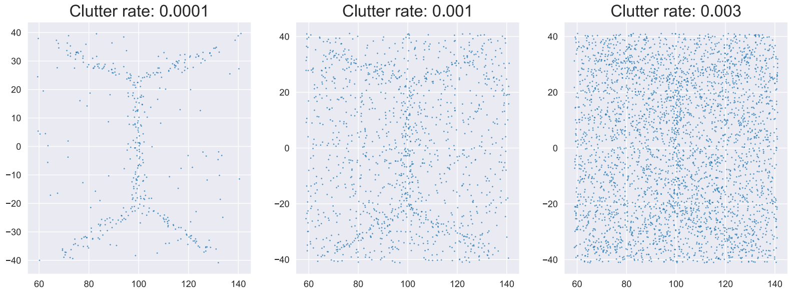


Figure 4.3: Data from simulator trough all time scans for different parameter λ.

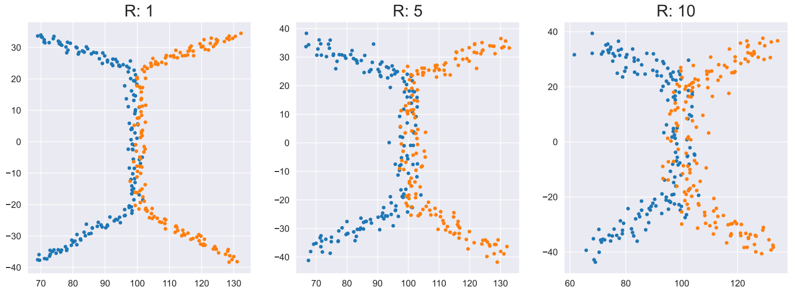


Figure 4.4: Trajectories for different $\mathbf{R}$.

Root Mean Square Error (RMSE) and Generalized Optimal Sub-Pattern Assignment (GOSPA) metric [28].

RMSE is a commonly used metric for evaluating the precision of state estimation. It measures the average Euclidean distance between the estimated target positions and their true positions. For a set of N targets and T time steps, the RMSE is calculated as:

$$RMSE = \sqrt{\frac{1}{NT} \sum_{n=1}^{N} \sum_{t=1}^{T} ||\hat{\mathbf{x}}_n^t - \mathbf{x}_n^t||^2}, \tag{4.2}$$

where $\hat{\mathbf{x}}_n^t$ and $\mathbf{x}_n^t$ represent the estimated and true positions of target n at time step t, respectively.

Although RMSE provides a simple measure of estimation accuracy, it does not account for errors in the number of tracks or data association mistakes. This limitation makes it insufficient for comprehensive evaluation of multi-target tracking algorithms, especially in scenarios prone to track coalescence.

The generalized optimal sub-pattern assignment or GOSPA metric [28] overcomes the limitations of RMSE by considering both localization errors and cardinality errors (missed and false tracks) in a unified framework. This is achieved by finding the optimal assignment between the set of true target states and the set of estimated tracks, considering penalties for missed targets, false tracks, and localization errors. The GOSPA metric is defined as follows:

$$GOSPA(X, Y) = \left(\min_{\pi \in \Pi_p} \sum_{i=1}^{m} d_c^{(c)}(x_i, y_{\pi(i)})^p + \frac{c^p}{p}(n - m) \right)^{1/p}, \tag{4.3}$$

where:

- $X = x_1, ..., x_m$ is the set of true target states.
- $Y = y_1, ..., y_n$ is the set of estimated tracks.
- m and n are the number of true targets and estimated tracks, respectively.
- Π_p is the set of all possible assignments between X and a subset of Y with at most p elements.
- $d_c^{(c)}(x, y) = \min(c, d(x, y))$ is the cut-off distance function, where $d(x, y)$ is the distance between states x and y, and c is a cut-off parameter.

- p determines the penalty for cardinality errors relative to localization errors.

It is necessary to define the parameters c and p. The GOSPA metric provides a more comprehensive evaluation of tracking performance compared to RMSE, making it suitable for analyzing the behavior of JPDA-based algorithms and their ability to handle track coalescence.

The final metric to be monitored is the number of misdetections reported by the algorithm. This could occur when the validation gate for a particular target is empty, causing the algorithm to omit the update step and declare a misdetection. Given that GOSPA conducts its own track-to-measurement assignment, the reported misdetections by the algorithm may vary from the cardinality errors identified by GOSPA.

4.2 Parameter Optimization via Monte Carlo Simulations

To optimize the performance of our method, we employ Monte Carlo simulations coupled with a grid search approach. This process involves evaluating the algorithm's performance across a range of parameter combinations and selecting the set that yields the best results. We optimize the following parameters.

- `P_squeeze`: The shrunken probability for the validation gate applied during potential track intersection scenarios.
- `squeeze_max_steps`: The maximum number of consecutive time steps for which the squeezed validation gate is applied.
- `cooldown_steps`: The number of time steps during which the standard validation gate is used after reaching the `squeeze_max_steps` limit.

For each combination of parameter values within the defined grid, we run the algorithm 100 times on the chosen trajectory. This allows us to account for the stochastic nature of the tracking process and obtain statistically meaningful results.

Before running simulations, it is essential to establish the simulation parameters, specifically, clutter rate λ, measurement uncertainty $\mathbf{R}$, detection and gating probability P_D and P_G, the type of trajectory, and the critical distance between tracks denoted by `mirror_offset`. The conditions should neither be overly challenging for the filter nor too simplistic. Through empirical methods, we have identified the parameters under which the filter encounters some challenges but does not immediately lose track. These parameters are as follows.

- $\lambda = 0.0015$,
- $\mathbf{R} = 5\mathbf{I}$,
- Trajectory type – **trapezoid**,
- `mirror_offset` $= 2$,
- $P_D = 1$,
- $P_G = 0.99$

The last thing to determine for this scenario are the GOSPA parameters: cut-off distance c and penalization term p. Cut-off distance represents the maximum distance beyond which two points are considered completely dissimilar or as a misdetection. Normally, it would depend on the physical properties of the given radar system and its resolution. Furthermore, if targets are known to be well-separated, a smaller c can be used to emphasize localization accuracy. Nonetheless, a smaller c could result in a uniform metric result, since the minimum is taken between the distance and c. The penalization term p controls the sensitivity of the metric to outliers and the relative weighting of cardinality and localization errors. Since all of the simulated tracks are moving with constant velocity and have predictable trajectory, we expect the predictions to be quite accurate, co there is no need to set c some high value.

In contrast, as discussed in Section 3.5, adaptive gating in our approach could lead to a higher incidence of misdetections as a consequence of decreasing P_G. This leads to the increased value of the penalization term in order to reduce the misdetection rate. We set

- $c = 20$,
- $p = 2$.

It is important to note that the optimal parameters may vary depending on the specific characteristics of the tracking scenario, including target dynamics, clutter density, and sensor capabilities. As such, the parameter optimization process should be tailored to the specific application context.

Results

The plots presented in Figure 4.5 illustrate the relationship between the adaptive gating parameters (`P_squeeze`, `squeeze_max_steps`, `cooldown`) and the performance metrics RMSE and GOSPA. First, let us look at `P_squeeze`. As `P_squeeze` increases (larger validation gate), both RMSE and GOSPA exhibit a general upward trend. This suggests that a more permissive gating approach leads to increased localization errors (RMSE) and potentially more cardinality errors (missed detections and false tracks reflected in GOSPA). This is likely due to the inclusion of more clutter and potentially incorrect data associations within the larger gate. The trade-off between localization accuracy and cardinality errors remains a key consideration.

With increasing `squeeze_max_steps` (longer duration of the shrunken gate), RMSE shows a sharp decrease, indicating improved localization accuracy due to a more focused validation region during potential track interactions. GOSPA demonstrates a similar behavior.

Both RMSE and GOSPA show relatively stable trends with increasing `cooldown` periods. This suggests that the `cooldown` duration has a less significant impact on overall tracking performance compared to `P_squeeze` and `squeeze_max_steps`.

Figure 4.6 displays the average number of misdetections. The number of

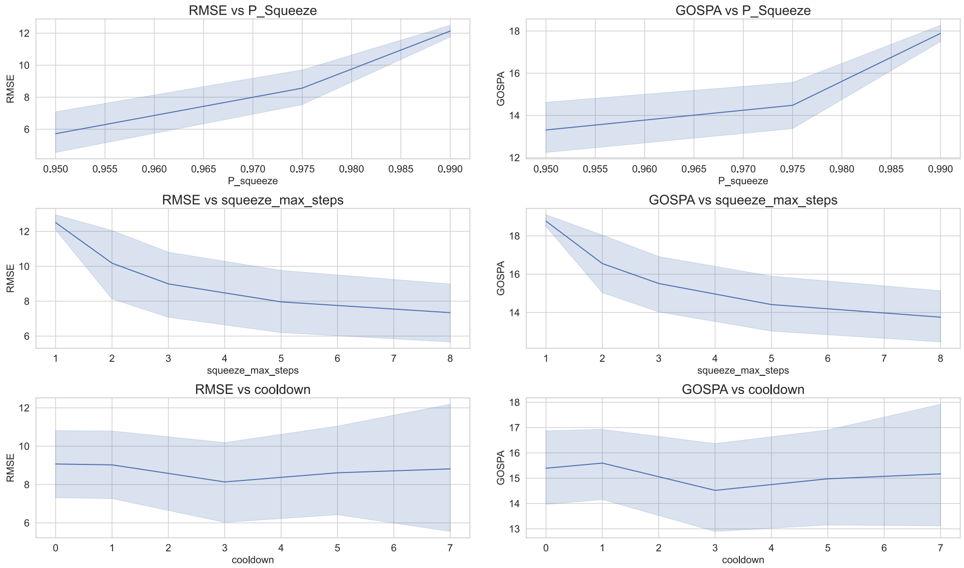


Figure 4.5: RMSE and GOSPA for different parameters.

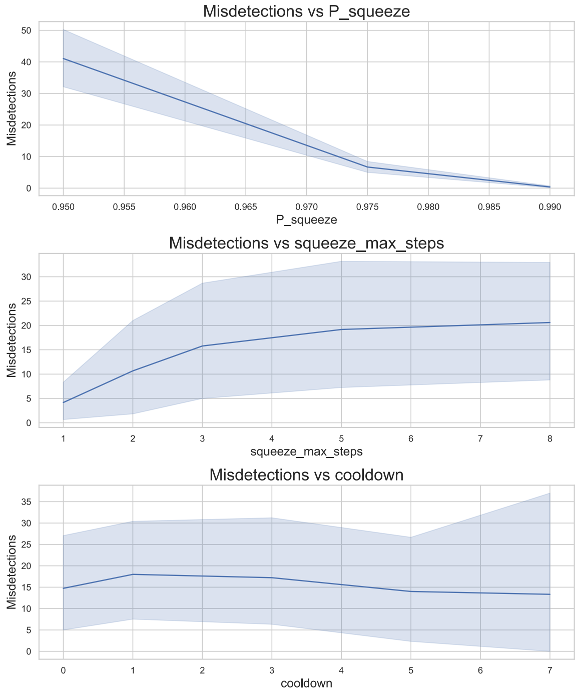


Figure 4.6: Misdetections for different parameters.

misdetections decreases significantly with increasing `P_squeeze`. Surprisingly, this does not correlate with the expected GOSPA trends. It is important to recognize that while GOSPA considers cardinality errors (including missed detections), it may not directly reflect the exact number of misdetections reported by the algorithm. This is due to several key factors:

- **Independent Association** GOSPA performs its own optimal association between true targets and estimated tracks, which may differ from the associations made by the tracking algorithm itself. As a result, GOSPA might identify potential associations between tracks and measurements that the algorithm classified as misdetections, leading to a discrepancy in the reported misdetection counts.
- **Constant Validation Gate** GOSPA operates with a fixed cut-off distance c for determining association possibilities, whereas the adaptive gating mechanism dynamically adjusts the size of the validation gate. This difference in gating strategies can further contribute to discrepancies between GOSPA's assessment of cardinality errors and the actual misdetections encountered by the algorithm.

However, Figure 4.7 clearly illustrates that both RMSE and GOSPA exhibit

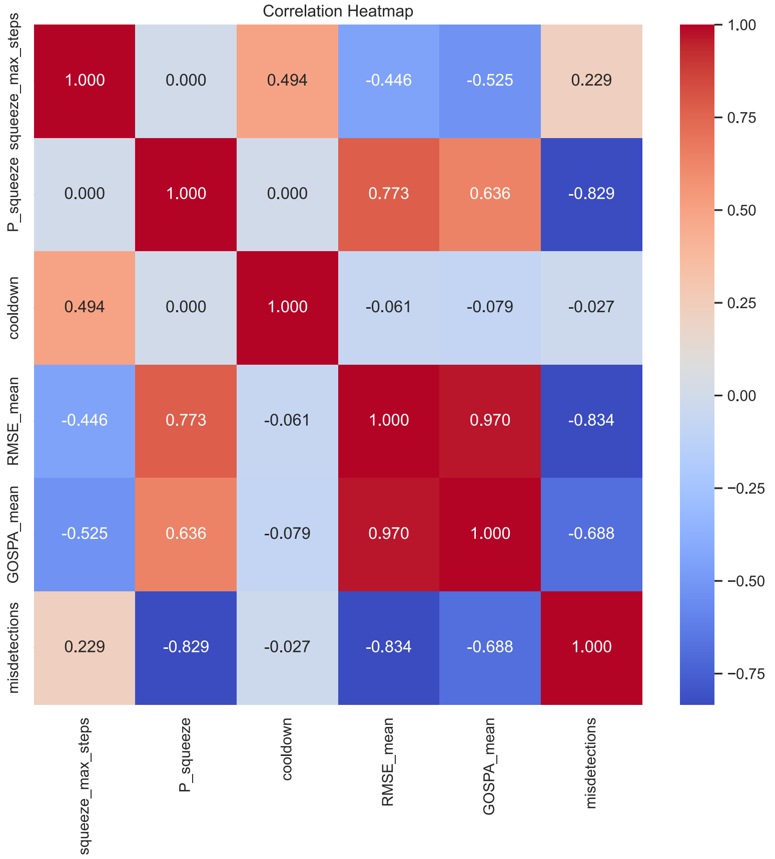


Figure 4.7: Correlation heatmap.

a strong negative correlation with the number of misdetections, although the correlation strength is lower for GOSPA (-0.688 compared to -0.834 for RMSE). This indicates that while GOSPA does account for cardinality errors to some extent, their influence on the overall metric is minor relative to localization errors.

Misdetections show a clear upward trend with increasing `squeeze_max_steps`, confirming that longer durations of the shrunken gate lead to a higher likelihood of missed detections. This is likely due to the exclusion of valid measurements when the gate is too restrictive for extended periods. The `cooldown` parameter appears to have minimal impact on the frequency of misdetections.

A correlation heatmap depicted in Figure 4.7 reinforces the observed trends. Both RMSE and GOSPA are strongly positively correlated with `P_squeeze` and negatively correlated with `squeeze_max_steps`. The parameter `cooldown` has a negligible negative correlation with the metrics. The relationship between misdetections and algorithm parameters is consistent with previous findings, emphasizing that a higher `P_squeeze` leads to fewer misdetections, and to a lesser degree, reducing `squeeze_max_steps` also contributes to fewer missed targets.

Based on these observations, selecting the optimal parameter combination involves finding a balance between minimizing localization errors (RMSE) and cardinality errors (GOSPA + misdetections). The specific choice will depend on the application priorities and the relative importance of accurate state estimation versus correct track maintenance. For example, if minimizing localization errors is crucial, a lower `P_squeeze` value and a higher `squeeze_max_steps` might be preferred. Conversely, if ensuring correct track cardinality is of greater importance, a higher `P_squeeze` and a shorter `squeeze_max_steps` might be more suitable, accepting a slight increase in localization errors. The `cooldown` parameter appears to have a less significant impact on performance and could be chosen based on other application-specific considerations.

For further simulations, we set

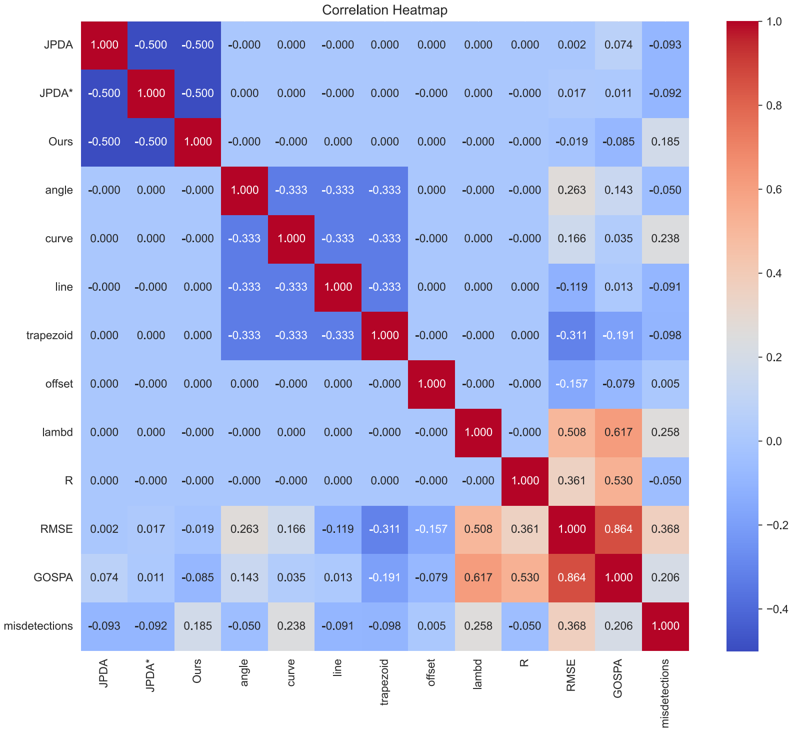
- `P_squeeze` $= 0.975$.
- `squeeze_max_steps` $= 3$.
- `cooldown` $= 2$

4.3 Algorithm Comparison: Controlled Experiment with Fixed Seeds

Once we have defined the optimal parameters of our method for a given scenario, we can compare it with other methods. To provide a controlled comparison of the JPDA, JPDA*, and our method, we conduct an experiment where we eliminate the stochastic factor by fixing the random seed during simulation. This allows us to isolate the impact of algorithm design and parameter choices on tracking performance.

This experiment is also based on grid search. The parameters for this grid search are

- `trajectory_type` $\leftarrow$ {angle, curve, line, trapezoid} (see Figure 4.1).
- `mirror_offset` $\leftarrow \{1,\ 2,\ 5,\ 10,\ 20\}$.



	JPDA	JPDA*	Ours	angle	curve	line	trapezoid	offset	lambd	R	RMSE	GOSPA	misdetections
JPDA	1.000	-0.500	-0.500	-0.000	0.000	-0.000	0.000	0.000	0.000	0.000	0.002	0.074	-0.093
JPDA*	-0.500	1.000	-0.500	0.000	0.000	-0.000	0.000	-0.000	0.000	-0.000	0.017	0.011	-0.092
Ours	-0.500	-0.500	1.000	-0.000	-0.000	0.000	0.000	0.000	-0.000	-0.000	-0.019	-0.085	0.185
angle	-0.000	0.000	-0.000	1.000	-0.333	-0.333	-0.333	0.000	-0.000	-0.000	0.263	0.143	-0.050
curve	0.000	0.000	-0.000	-0.333	1.000	-0.333	-0.333	-0.000	0.000	-0.000	0.166	0.035	0.238
line	-0.000	-0.000	0.000	-0.333	-0.333	1.000	-0.333	0.000	0.000	0.000	-0.119	0.013	-0.091
trapezoid	0.000	0.000	0.000	-0.333	-0.333	-0.333	1.000	-0.000	-0.000	0.000	-0.311	-0.191	-0.098
offset	0.000	-0.000	0.000	0.000	-0.000	0.000	-0.000	1.000	-0.000	-0.000	-0.157	-0.079	0.005
lambd	0.000	0.000	-0.000	-0.000	0.000	0.000	-0.000	-0.000	1.000	-0.000	0.508	0.617	0.258
R	0.000	-0.000	-0.000	-0.000	-0.000	0.000	0.000	-0.000	-0.000	1.000	0.361	0.530	-0.050
RMSE	0.002	0.017	-0.019	0.263	0.166	-0.119	-0.311	-0.157	0.508	0.361	1.000	0.864	0.368
GOSPA	0.074	0.011	-0.085	0.143	0.035	0.013	-0.191	-0.079	0.617	0.530	0.864	1.000	0.206
misdetections	-0.093	-0.092	0.185	-0.050	0.238	-0.091	-0.098	0.005	0.258	-0.050	0.368	0.206	1.000

Figure 4.8: Correlation heatmap.

- $\lambda \leftarrow \{0.0001,\ 0.0005, 0.001, 0.003\}$.
- $\mathbf{R} \leftarrow \{0.5,\ 1,\ 5,\ 10\}$.

This gives us 240 combinations in total for each algorithm.

Results

Figure 4.8 displays a correlation heatmap that features the experiment parameters. The heatmap reveals intriguing relationships between algorithm performance (RMSE and GOSPA) and the different scenario parameters. RMSE shows a moderate positive correlation with λ (clutter rate) and $\mathbf{R}$ (measurement uncertainty). This is expected, as higher clutter density and measurement noise generally lead to increased localization errors for all algorithms. The correlation is stronger with λ. GOSPA exhibits a stronger positive correlation with $\mathbf{R}$ than RMSE. Otherwise, we see a similar dependence. Surprisingly, RMSE shows a stronger positive correlation with the amount of misdetections, which is more expected from GOSPA.

Interestingly, both RMSE and GOSPA show weak negative correlations with the `mirror_offset` parameter. This suggests that as the distance between tracks increases, the performance of all algorithms tends to improve slightly. In particular, this performance enhancement is noticeable for trajectories *line* and *trapezoid*, where targets move along a straight line in close proximity. Under such conditions with an increased clutter rate λ and $\mathbf{R}$ tracking becomes particularly impossible (refer to Figure 4.4 where $\mathbf{R} = 10$).

However, in Figure 4.8, we can clearly see that the *angle* trajectory has the strongest positive correlation with RMSE and GOSPA among other trajectories. It could be caused by a sharp turn in the trajectory at a critical point. The implemented algorithms use CVM as a motion model, which is not well suited for such rapid changes in motion. The most complicated trajectory in terms of misdetections is *curve*, which is expected. Again, the reason is the constant velocity motion model and the highly maneuvering nature of the trajectory.

The heatmap also provides subtle hints about the relative performance of the algorithms. The correlation patterns for baseline JPDA, JPDA*, and our proposed method appear quite similar. This indicates that overall performance trends with respect to the scenario parameters are consistent across all three algorithms. However, slight differences in the magnitude of the correlations can be observed. For example, the negative correlation for GOSPA and RMSE seems slightly stronger for our proposed method compared to JPDA and JPDA* with pruning, both of which have a low positive correlation. This could imply that our approach potentially exhibits improved performance in various tracking scenarios.

Figure 4.9 displays three boxplots representing the metrics for three different algorithms. We see that JPDA* shows an increase in mean RMSE compared to baseline JPDA. However, we can see that the median RMSE for JPDA* is 32.7% lower than for the baseline JPDA, which means that the error of JPDA* might be higher in extreme cases, but remains pretty low in normal conditions. Our method achieves the lowest mean and median RMSE values, indicating the best localization performance among the three algorithms. This implies that the adaptive gating mechanism further reduces the impact of track coalescence on state estimation, resulting in more accurate target tracking.

The baseline JPDA shows the highest median and mean GOSPA values. This suggests that the baseline JPDA suffers from a combination of localization errors and cardinality errors (missed detections and false tracks), leading to suboptimal overall tracking performance. JPDA* exhibits a slight improvement in GOSPA compared to baseline JPDA, with lower mean and median values. This indicates that pruning helps reduce cardinality errors to some

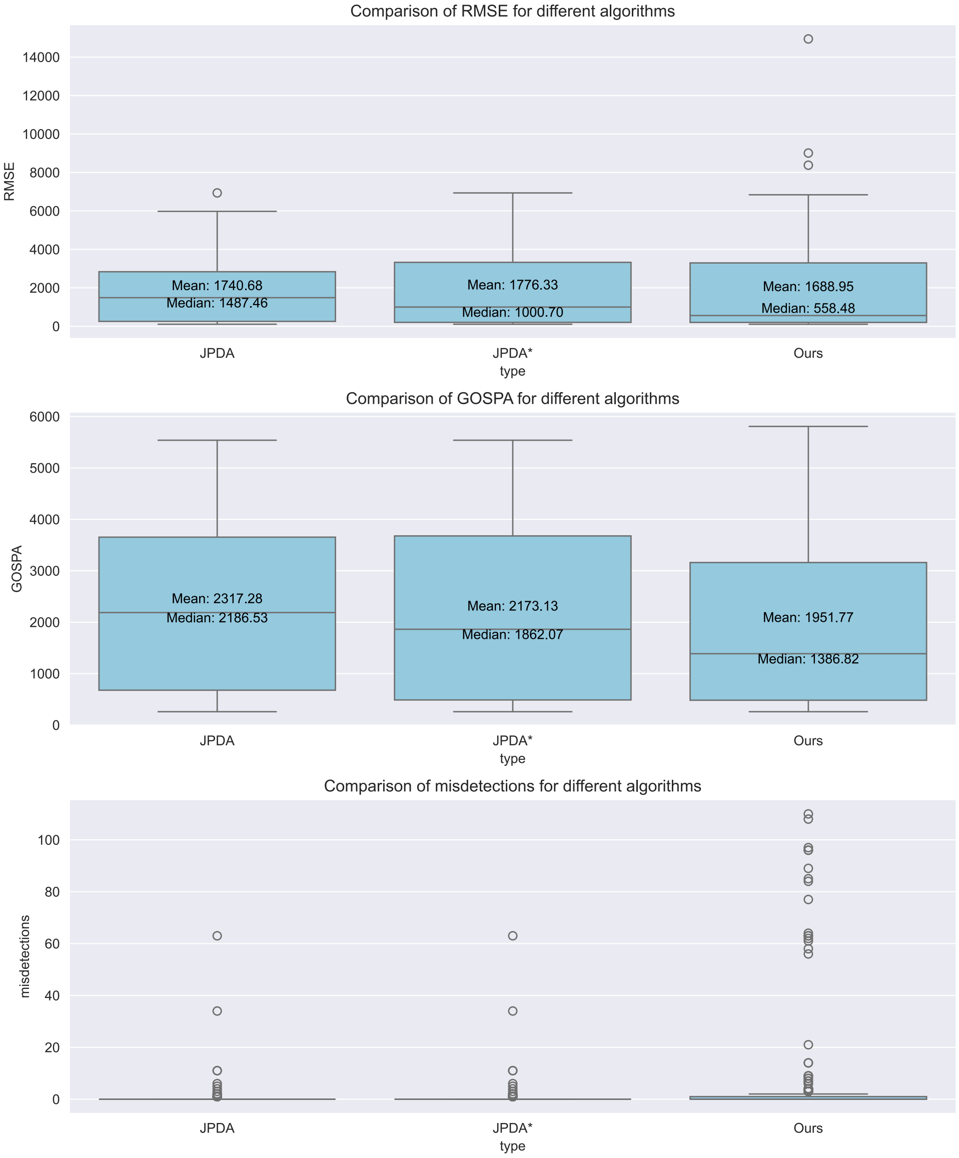


Figure 4.9: Metrics for JPDA, JPDA* and our method.

extent, but the improvement is not as significant as the reduction in RMSE. The proposed method demonstrates the lowest median and mean GOSPA values, which represents the best overall tracking performance among the three algorithms. This implies that the adaptive gating approach effectively balances localization accuracy and cardinality errors, leading to more reliable and accurate multi-target tracking.

However, we can see an increase in the number of misdetections for our approach, compared to JPDA and JPDA*. Both show similar performance in terms of misdetections. In Table 4.1 we can see that baseline JPDA and JPDA* exhibit very low misdetection rates across all percentiles. The median and 75th percentile values of 0 indicate that in most scenarios, these algorithms experience no misdetections. Even at the 90th percentile, the number of misdetections remains low, 1 for both algorithms. This confirms their effectiveness in maintaining track continuity and avoiding track loss. Our proposed method with adaptive gating, while still having a median misdetection value of 0, shows a significantly higher number of misdetections at higher percentiles. The 75th percentile value of 1 suggests that in a quarter of the scenarios, our method experiences at least one misdetection. This increases to 4 misdetections at the 90th percentile, indicating a higher susceptibility to missed detections in certain challenging situations. The data suggest that the increased misdetection rate in our proposed method is not a uniform issue across all scenarios but rather concentrated in a subset of challenging situations. While most scenarios still result in no misdetections, specific conditions lead to a higher number of missed tracks.

	JPDA	JPDA*	Ours
Mean	0.52	0.53	4.81
Median	0	0	0
75th percentile	0	0	1
90th percentile	1	1	4

Table 4.1: Mean Misdetections for Three Algorithms

4.4 Direct Algorithm Comparison in Specific Scenarios

To further evaluate the performance of the JPDA, JPDA*, and our proposed method with adaptive gating, we conduct a direct comparison in three concrete tracking scenarios with varying levels of difficulty. This allows us to assess how each algorithm handles specific challenges and observe their behavior under controlled conditions.

We define three scenarios with different parameter settings to represent *easy*, *medium*, and *hard* tracking conditions. These scenarios are designed to vary the levels of measurement uncertainty, clutter density, and target proximity, which are key factors that influence track coalescence and overall tracking performance. The values of the specific parameters for each scenario are presented in Table 4.2.

Parameter	Easy	Medium	Hard
$\mathbf{R}$	1.0	1.5	2.0
P_D	1.0	1.0	1.0
P_G	0.999	0.999	0.999
Trajectory	Trapezoid	Trapezoid	Trapezoid
Offset	5	3	3
λ	0.001	0.0015	0.003
$P_{squeeze}$	0.975	0.975	0.975
Steps/Cooldown	3/2	3/2	3/2

Table 4.2: Parameter settings for the easy, medium, and hard tracking scenarios.

As shown in the table, the scenarios primarily differ in the following aspects:

- **Measurement Uncertainty** ($\mathbf{R}$): The measurement noise covariance R increases from easy to hard scenarios, representing an increase in uncertainty in sensor measurements.
- **Clutter Density** (λ): The clutter rate λ also increases from easy to hard scenarios, indicating a higher density of false alarms and background noise.
- **Target Proximity** (Offset): The distance between targets at their closest point (offset) is smaller in the medium and hard scenarios compared to the easy scenario, creating a higher risk of track coalescence due to closer target interactions.

By comparing the algorithms across these scenarios, we can gain insights into their robustness and ability to handle different levels of difficulty and specific tracking challenges.

The graphs showing RMSE and GOSPA over time for each scenario (Figures 4.10, 4.11, and 4.12) provide valuable information on the comparative performance and behavior of JPDA, JPDA*, and our proposed method with adaptive gating under varying levels of difficulty. Furthermore, Figure 4.13

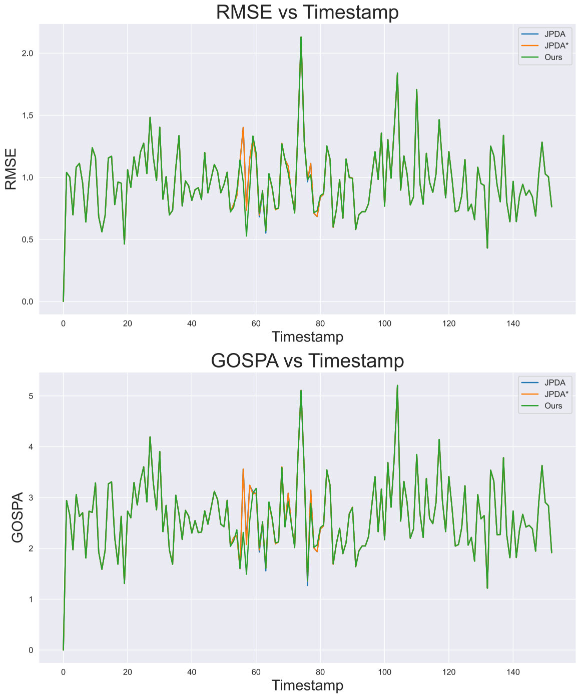


Figure 4.10: Metrics for *easy* scenario.

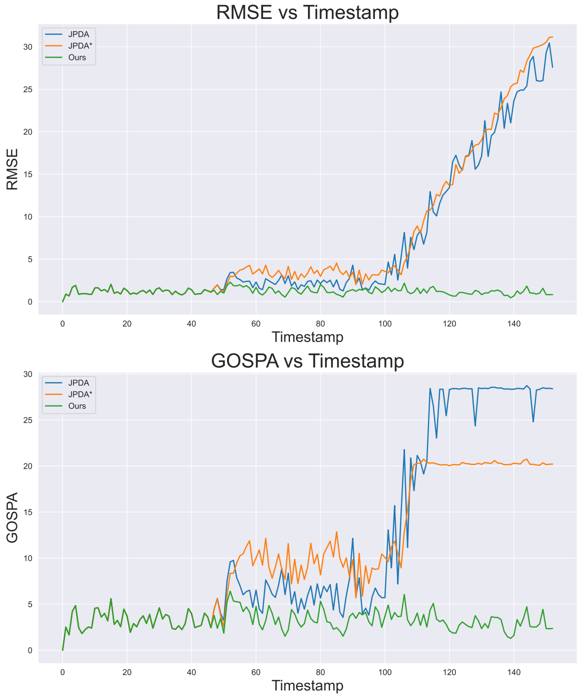


Figure 4.11: Metrics for *medium* scenario.

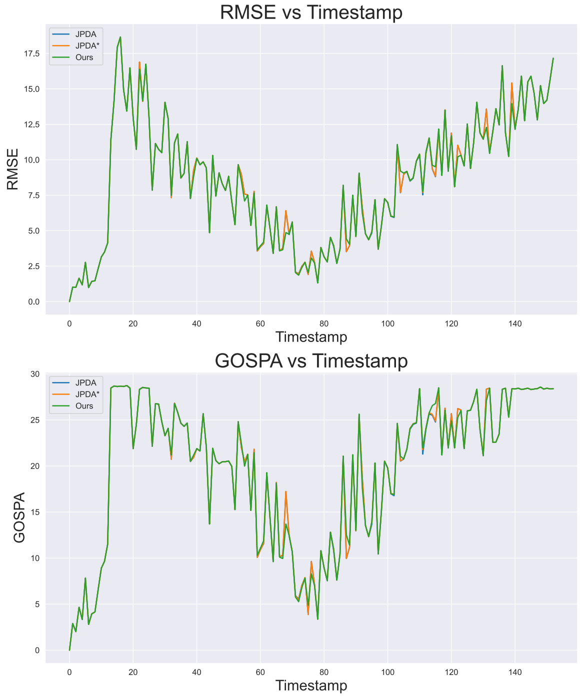
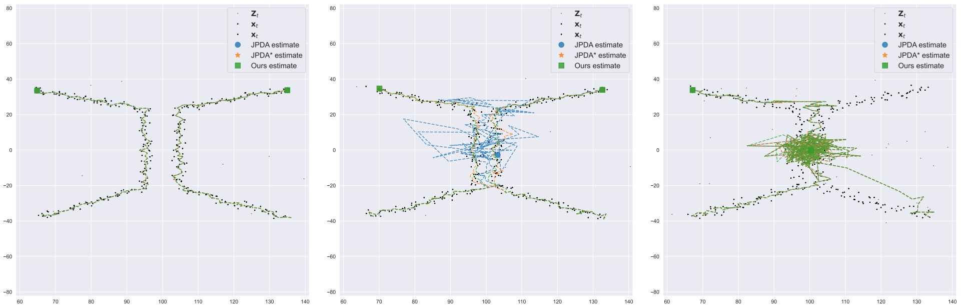


Figure 4.12: Metrics for *hard* scenario.



(a) *Easy* scenario.

(b) *Medium* scenario.

(c) *Hard* scenario.

Figure 4.13: Trajectories for different scenarios at time $t = 153$.

shows the trajectories generated by each algorithm in all three scenarios. Let us analyze the observations for each scenario.

All three algorithms exhibit similar performance in terms of both RMSE and GOSPA for the *easy* scenario (see Figure 4.10). The metrics remain relatively stable and low throughout the scenario, indicating successful tracking of both targets with minimal localization and cardinality errors. This suggests that under certain tracking conditions with low measurement noise, clutter density, and sufficient target separation, all algorithms can effectively maintain track accuracy and avoid track coalescence.

Concerning the *medium* scenario, we observe the following.

- JPDA and JPDA* with pruning show a significant increase in RMSE compared to the easy scenario, particularly after the point where the target trajectories approach each other. This indicates that both algorithms struggle with increased measurement uncertainty, clutter, and reduced target separation, leading to track loss.
- Our proposed method with adaptive gating demonstrates significantly lower RMSE and GOSPA throughout the scenario, including during the critical track interaction phase. This highlights the effectiveness of adaptive gating in mitigating track coalescence and maintaining accurate state estimation even under more challenging conditions.
- GOSPA trends for all algorithms follow a similar pattern to the RMSE observations, further confirming the superior performance of our proposed method in terms of overall tracking accuracy and cardinality error management. Furthermore, Figure 4.11 illustrates the difference in GOSPA between baseline JPDA and JPDA* during the last scans, while RMSE shows similar behavior. It indicates fewer cardinality errors for JPDA*.

For the *hard* scenario, we see that all three algorithms experience a rapid increase in both RMSE and GOSPA shortly after the start of the scenario, indicating a loss of track accuracy and increased cardinality errors. The high level of measurement noise, clutter density, and close target proximity pose significant challenges for all algorithms, leading to track coalescence and ultimately track loss. The observed decrease in RMSE during the period of closest target proximity is likely an artifact of the evaluation metric, rather than an indication of improved tracking performance. In Figure 4.13, we can see that when the filter loses track, its predictions are concentrated in the center of the surveillance area. As the true target states approach each other in the center, even erroneous state estimates would result in lower RMSE values due to the reduced distance between the estimated and true positions. The subsequent increase in RMSE and GOSPA as the targets move apart further confirms the loss of track accuracy and the presence of significant cardinality errors.

The results across the three scenarios clearly demonstrate the advantages of our proposed JPDA* variant with adaptive gating in handling challenging tracking conditions. Although all algorithms perform well in the *easy* scenario, our method exhibits superior robustness and accuracy in the *medium* scenario, successfully mitigating track coalescence and maintaining reliable tracking performance. For a *hard* scenario, our algorithm fails, as well as other ones. Concerning misdetections, no algorithm reported any misdetections in these scenarios.

These findings highlight the effectiveness of adaptive gating as a valuable technique to enhance JPDA-based tracking algorithms, particularly in scenarios with increased, but still manageable, measurement uncertainty, clutter density, and close target interactions. The ability to dynamically adjust the validation gate based on target proximity and estimated states enables our method to effectively balance the trade-off between localization accuracy and cardinality errors, leading to more robust and reliable multi-target tracking performance in complex and challenging environments. In extreme situations, however, our algorithm shows a higher degree of misdetections than others, which could be a foundation for future work.

Conclusion and Future Work

This thesis has explored the challenges and solutions associated with track coalescence in Joint Probabilistic Data Association (JPDA) filters, a widely used class of multi-target tracking algorithms. We began by establishing a comprehensive understanding of single-target tracking (STT) principles, including state estimation, motion models, and the impact of measurement noise and clutter. The limitations of traditional Kalman filters in handling clutter led to the introduction of the Probabilistic Data Association (PDA) filter, which effectively incorporates data association probabilities to mitigate the adverse effects of clutter and missed detections.

Moving into the realm of multi-target tracking (MTT), we delved into the complexities of tracking multiple objects simultaneously and the critical issue of track coalescence. JPDA filters, as an extension of PDA for MTT scenarios, were analyzed in detail, highlighting their vulnerability to track coalescence when targets move in close proximity. The factors contributing to track coalescence, including high clutter density, low detection probability, and similar target motion, were discussed, along with the detrimental consequences of track loss, degraded state estimation, and inaccurate target identification.

To address the challenge of track coalescence, we investigated various existing suppression techniques, including:

- **Exact Nearest Neighbor (ENNPDA)** A simple yet effective approach that prunes all but the most likely association hypothesis, leading to reduced coalescence but increased sensitivity to clutter and misdetections.
- **JPDA*** A more robust approach that retains the clutter and misdetection resilience of JPDA while strategically pruning less likely hypotheses

to mitigate coalescence.

- **Coupling Methods (e.g., CPDA, CPDA*)** Techniques that utilize a joint state representation for multiple targets, offering some improvement in handling coalescence, but often with increased complexity.
- **Bias Removal Methods (e.g., BRJPDA)** Approaches that estimate and remove bias from the state estimates to counteract the tendency of tracks to merge during close interactions.

Building upon the concept of hypothesis pruning in JPDA*, we introduced a novel method that incorporates adaptive validation gating to further enhance coalescence suppression. Our proposed algorithm dynamically adjusts the validation gate size based on the proximity and estimated states of nearby targets. By shrinking the gate during potential track interactions, we aim to limit association possibilities and reduce the likelihood of tracks merging erroneously.

To evaluate the performance of our proposed method, we conducted simulations and experiments.

- **Monte Carlo Simulations** A grid search approach was employed to identify the optimal parameter settings for our adaptive gating mechanism, considering the trade-offs between localization accuracy, cardinality errors, and misdetections.
- **Controlled Experiment with Fixed Seeds** A comprehensive comparison of JPDA, JPDA*, and our proposed method was conducted across a wide range of scenarios with varying clutter densities, measurement uncertainties, target proximities, and trajectory types. The results provided insights into the relative performance and robustness of each algorithm under different conditions.
- **Direct Comparison in Specific Scenarios** We directly compared the algorithms in three concrete scenarios that represent easy, medium and hard tracking conditions. This allowed us to assess their behavior and effectiveness in handling specific challenges.

The experimental results demonstrate that our proposed JPDA* variant with adaptive gating generally outperforms both the baseline JPDA and JPDA* with pruning. Our method consistently achieves:

- **Improved Localization Accuracy** The adaptive gating mechanism effectively mitigates track coalescence, leading to more accurate state estimation and reduced localization errors (RMSE) compared to the other algorithms, especially in scenarios with moderate difficulty.
- **Enhanced Overall Tracking Performance** By balancing the trade-off between localization and cardinality errors, our method achieves the best overall tracking performance (GOSPA) among the algorithms tested.

However, our approach exhibits a drawback in terms of a higher misdetection rate in extreme or challenging scenarios. While the median number of misdetections remains low, indicating reliable track maintenance in most situations, specific conditions can lead to increased missed detections due to the potentially aggressive nature of adaptive gating.

This limitation paves the way for future research and improvements.

- **Adaptive `P_squeeze`**: Our current implementation uses a predefined value for `P_squeeze`, the probability used to shrink the validation gate. To further enhance the adaptability of the algorithm, `P_squeeze` could be dynamically adjusted based on factors such as:
  - **Target Proximity** A more nuanced approach could scale `P_squeeze` based on the distance between intersecting gates or the estimated distance between targets, allowing for a more sophisticated control of the gate size.
  - **Track Quality Metrics** `P_squeeze` could be adapted based on estimated track quality or confidence level, providing a more informed gating decision that considers the reliability of the tracks involved.
  - **Clutter Density and Measurement Noise** The gating probability could be dynamically adjusted to account for varying levels of clutter and measurement noise, ensuring a more robust response to changing environmental conditions.
- **Alternative Gating Strategies** Exploring alternative gating shapes, such as elliptical or rectangular gates with dynamically adjusted dimensions, could offer additional flexibility and potentially improve performance in specific scenarios.
- **Advanced Track Management** Integrating more sophisticated track management strategies that consider track quality, age, and the like-

lihood of misdetections could further enhance track maintenance and reduce the impact of missed detections on overall tracking performance.

- **Integration with Machine Learning:** Incorporating machine learning techniques to learn optimal gating parameters or predict track coalescence events could lead to more intelligent and adaptive tracking systems.

By addressing the limitations of our proposed method and exploring these avenues for future research, we can strive towards even more robust, accurate, and adaptable multi-target tracking solutions capable of effectively handling complex and challenging scenarios with closely spaced targets.

Bibliography

[1] Rezatofighi, S. H.; Gould, S.; et al. Multi-target tracking with time-varying clutter rate and detection profile: application to time-lapse cell microscopy sequences. *IEEE Transactions on Medical Imaging*, volume 34, no. 6, Jan. 2015: pp. 1336–1348.

[2] Chen, X.; Zhou, X.; et al. Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy. *IEEE transactions on biomedical engineering*, volume 53, no. 4, Mar. 2006: pp. 762–766.

[3] Choi, J.; Ulbrich, S.; et al. Multi-Target Tracking using a 3D-Lidar sensor for autonomous vehicles. In *16th International IEEE Conference on Intelligent Transportation Systems*, Oct. 2013, pp. 881 – 886.

[4] Lopez, R.; Malardé, J.-P.; et al. Improving Argos Doppler Location Using Multiple-Model Kalman Filtering. *IEEE Transactions on Geoscience and Remote Sensing*, volume 52, no. 8, Oct. 2014: pp. 4744–4755.

[5] Blackman, S. S.; Broida, T. J. Multiple sensor data association and fusion in aerospace applications. *Journal of Robotic Systems*, volume 7, no. 3, June 1990: pp. 445–485.

[6] Benfold, B.; Reid, I. Stable multi-target tracking in real-time surveillance video. In *CVPR 2011*, June 2011, pp. 3457–3464.

[7] Kalman, R. E. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, volume 82, Mar. 1960: pp. 35–45.

[8] Del Moral, P. Nonlinear filtering: interacting particle resolution. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, volume 325, no. 6, Sept. 1997: pp. 653–658.

[9] Wan, E.; Van Der Merwe, R. The unscented Kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, Oct. 2000, pp. 153–158.

[10] Rong Li, X.; Bar-Shalom, Y. Tracking in clutter with nearest neighbor filters: analysis and performance. *IEEE Transactions on Aerospace and Electronic Systems*, volume 32, no. 3, July 1996: pp. 995–1010.

[11] Bar-Shalom, Y.; Tse, E. Tracking in a cluttered environment with probabilistic data association. *Automatica*, volume 11, no. 5, Sept. 1975: pp. 451–460.

[12] Reid, D. An Algorithm for Tracking Multiple Targets. *IEEE Transactions on Automatic Control*, volume 24, no. 6, Dec. 1979: pp. 843–854.

[13] Bar-Shalom, Y.; Daum, F.; et al. The probabilistic data association filter. *IEEE Control Systems*, volume 29, no. 6, Dec. 2009: pp. 82–100.

[14] Vo, B.; Ma, W. The gaussian mixture probability hypothesis density filter. *IEEE Transactions on Signal Processing*, volume 54, no. 11, Nov. 2006: pp. 4091–4104.

[15] Fitzgerald, R. J. Development of practical PDA logic for multitarget tracking by microprocessor. In *1986 American Control Conference*, June 1986, pp. 889–898.

[16] Blom, H.; Bloem, E. Probabilistic data association avoiding track coalescence. *IEEE Transactions on Automatic Control*, volume 45, no. 2, Feb. 2000: pp. 247–259.

[17] Bar-Shalom, Y.; Li, X.-R. *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS publishing Storrs, CT, 1995.

[18] Brekke, E. *Fundamentals of Sensor Fusion: Target Tracking, Navigation and Slam*. Norwegian University of Science and Technology, 2020.

[19] Kosuge, Y.; Matsuzaki, T. The optimum gate shape and threshold for target tracking. In *SICE 2003 Annual Conference*, volume 2, Aug. 2003, pp. 2152–2157.

[20] Willett, P.; Ruan, Y.; et al. PMHT: problems and some solutions. *IEEE Transactions on Aerospace and Electronic Systems*, volume 38, no. 3, July 2002: pp. 738–754.

[21] Mušicki, D.; Evans, R.; et al. Integrated Probabilistic Data Association. *IEEE Transactions on Automatic Control*, volume 39, no. 6, June 1994: pp. 1237–1241.

[22] Kropfreiter, T.; Meyer, F.; et al. To coalesce or to repel? An analysis of MHT, JPDA, and belief propagation multitarget tracking methods. *arXiv preprint arXiv:2308.06326*, Aug. 2023.

[23] Rong Li, X.; Bar-Shalom, Y. Tracking in clutter with nearest neighbor filters: analysis and performance. *IEEE Transactions on Aerospace and Electronic Systems*, volume 32, no. 3, July 1996: pp. 995–1010.

[24] Shertukde, H. M.; Bar-Shalom, Y. Tracking of crossing targets with imaging sensors. *IEEE transactions on aerospace and electronic systems*, volume 27, no. 4, July 1991: pp. 582–592.

[25] Bar-Shalom, Y.; Chang, K.; et al. Tracking of splitting targets in clutter using an interacting multiple model joint probabilistic data association filter. In *Proceedings of the 30th IEEE Conference on Decision and Control*, Dec. 1991, pp. 2043–2048.

[26] Fitzgerald, R. J. Track biases and coalescence with probabilistic data association. *IEEE Transactions on Aerospace and Electronic Systems*, volume AES-21, no. 6, Nov. 1985: pp. 822–825.

[27] Jing, P.; Xu, S.; et al. Coalescence-avoiding joint probabilistic data association based on bias removal. *EURASIP Journal on Advances in Signal Processing*, Mar. 2015: pp. 1–13.

[28] Rahmathullah, A. S.; García-Fernández, Á. F.; et al. Generalized Optimal Sub-Pattern Assignment Metric. In *2017 20th International Conference on Information Fusion*, IEEE, July 2017, pp. 1–8.

Appendix **A**

Acronyms

MTT Multi-target tracking

JPDA Joint Probabilistic Data Association

JIPDA Joint Integrated Probabilistic Data Association

RFS Random Finite Set

PHD Probability Hypothesis Density

PMBM Poisson multi-Bernoulli mixture

PDA Probability Data Association

MHT Multiple Hypothesis Tracking

GNN Global Nearest Neighbor

KF Kalman Filter

EKF Extended Kalman Filter

PF Particle Filter

UKF Unscented Kalman Filter

NN Nearest Neighbor

ENNPDA Exact Nearest Neighbor Probabilistic Data Association

CPDA Coupled Probabilistic Data Association

JPDAC Joint Probabilistic Data Association with Coupling

BRJPDA Bias Removal Joint Probabilistic Data Association

RMSE Root Mean Square Error

GOSPA Generalized Optimal Sub-Pattern Assignment

CDF Cumulative Distribution Function

i.i.d. independent and identically distributed

pdf probability density function

PPP Poisson Point Process

MMSE Minimum Mean Square Error

MAP Maximum A Posteriori

CVM Constant Velocity Model

CAM Constant Acceleration Model

Appendix B

Contents of enclosed media

```
env ................................ the directory with environment configs
 ├── environment.yml ....................... anaconda required packages
 └── gospapy ........................................... gospapy library
source ................................... the directory of source codes
thesis ............................................ the thesis directory
 ├── tex_src ........... the directory of LaTeX source codes of the thesis
 └── thesis.pdf ............................ the thesis text in PDF format
README.txt ......................... the file with repo contents description
```