FACULTY OF INFORMATION TECHNOLOGY CTU IN PRAGUE

Assignment of master's thesis

Title:	Distributed multiple target tracking
Student:	Bc. Jan Novák
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

Abstract: The state-of-the-art multiple target tracking (MTT) filters such as the (C)PHD or PMBM filters are mostly based on the random finite sets formulations. This brings high robustness and scalability. However, the price paid for this advantage is their enormous computational and memory burden. On the other hand, there are simpler less-demanding filters whose tracking performance is still very good. For instance, the (joint) probabilistic data association (J)PDA filters or the integrated probabilistic association filters (IPDA). It is possible to conjecture that if the simpler filters collaborate in a network, the tracking performance could be significantly improved while the computational and memory requirements remain low.

Goals:
1) Get familiar with the (J)PDA and/or IPDA filters. Study the initialization and deletion of tracks.
2) Study the possibilities of collaboration of several filters in a network.
3) Propose a method for collaboration among multiple MTT filters.
4) Perform an experimental validation of the resulting solution using simulated data.

Keywords: PDAF, JPDAF, multi-target tracking, data fusion, information fusion.

References:
[1] R. Repp, P. Rajmic, F. Meyer and F. Hlawatsch, "Target Tracking Using a Distributed Particle-Pda Filter With Sparsity-Promoting Likelihood Consensus," 2018 IEEE Statistical

---

Electronically approved by Ing. Magda Friedjungová, Ph.D. on 29 April 2023 in Prague.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Signal Processing Workshop (SSP), Freiburg im Breisgau, Germany, 2018, pp. 653-657, doi: 10.1109/SSP.2018.8450815.
[2] Y. Bar-Shalom, F. Daum and J. Huang, "The probabilistic data association filter," in IEEE Control Systems Magazine, vol. 29, no. 6, pp. 82-100, Dec. 2009, doi: 10.1109/MCS.2009.934469.
[3] E. Brekke, Fundamentals of Sensor Fusion. Norwegian University of Science and Technology, 2020.
[4] C. Fantacci, B. -N. Vo, B. -T. Vo, G. Battistelli and L. Chisci, "Robust Fusion for Multisensor Multiobject Tracking," in IEEE Signal Processing Letters, vol. 25, no. 5, pp. 640-644, May 2018, doi: 10.1109/LSP.2018.2811750.

Master's thesis

DISTRIBUTED MULTIPLE TARGET TRACKING

Bc. Jan Novák

Faculty of Information Technology
Department of Applied Mathematics
Supervisor: doc. Ing. Kamil Dedecius, Ph.D.
May 9, 2024

Contents

List of Figures

First and foremost, I would like to thank my supervisor, doc. Ing. Kamil Dedecius, Ph.D., for his neverending enthusiasm, encouragement, positive attitude, patience, and assistance in his classes and during supervision of this thesis.
I would also like to thank my family and girlfriend for their unwavering support during my studies. I would not stand where I am today without you.

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

Abstract

This thesis focuses on the topic of distributed multiple-target tracking. It presents the necessary mathematical concepts needed to understand the subject. Then, it introduces the concept of hidden state estimation with the hidden Markovian model and measurement model. Next, it introduces and derives the Kalman filter using the exponential family of distributions. The Kalman filter is then gradually evolved into the PDA filter, IPDA filter and JIPDA filter. Once these concepts are established, they deal with collaborative filtering methods, focusing mainly on the diffusion approach. The diffusion approach is initially described in a general manner. After that, a diffusion algorithm based on the JIPDA filter is proposed. The performance of the proposed algorithm is then evaluated in several experiments.

Keywords PDAF, JPDAF, single-target tracking, multi-target tracking, data fusion, information fusion, diffusion network, Kalman filter, Bayesian interference, Gaussian mixture

Abstrakt

Tato práce se zaměřuje na téma distribuovaného sledování více cílů. Nejprve popisuje nezbytné matematické koncepty potřebné k porozumění tématu. Poté představuje koncept odhadu skrytého stavu s pomocí skrytého Markovova modelu a modelu měření. Následně představuje a odvozuje Kalmanův filtr s využitím exponenciální rodiny distribucí. Z Kalmanova filtru jsou dále postupně odvozen PDA, IPDA, JIPDA a JIPDA filtry. Jakmile jsou tyto koncepty zavedeny, zabývá se metodami distribuovaného filtrování, přičemž se hlavně zaměřuje na difúzní přístup. Difúzní přístup je nejprve popsán obecným způsobem. Poté je navržen difúzní algoritmus založený na JIPDA filtru. Funkčnost navrženého algoritmu je pak vyhodnocena v několika experimentech.

Klíčová slova PDAF, JPDAF, sledování jednoho cíle, sledování více cílů, fúze dat, fúze informace, difúzní síť, Kalmanův filtr, Bayesovská inference, Gaussovská směs

List of abbreviations

CVM	constant velocity model
CAM	constant acceleration model
CPHD	cardinality probability hypothesis density
FOV	field of view
HPPP	homogenous Poisson point process
IPDA	integrated probability data association
JPDA	joint Probability data association
KF	Kalman filter
KLD	Kullback-Leilbler divergence
MHT	multiple hypotheses tracking
PDA	probability data association
PHD	probability hypothesis density
PMBM	Poisson multi-Bernoulli mixture
RFS	random finite set

Introduction

Ever since the development of the first radar installation before the 2nd World war, the development of target-tracking techniques has been mainly driven by military applications. The invention of long-range guided rockets has underlined the need to develop methods of identifying and tracking a target, with the aim of modeling and predicting its position in time to deploy appropriate countermeasures.

The results of the military research have later found applications in the civilian sector, notably, the first civilian application was air traffic control and maritime navigation aid, also known as automatic radar plotting aid (ARPA). The primary purpose of these systems was the very opposite of the military application that is, to prevent collisions of nearby airplanes or ships by keeping track of their whereabouts.

With the start of the space race arose the need to develop robust guidance systems, probes, and shuttles. This culminated in one of the most widely known examples of usage of the Kalman filter in the Apollo program as a part of its navigational and guidance computer. Later on, space agencies around the world started to employ target tracking algorithms to monitor the movement of space debris [1], asteroids in our solar system [2], or in guidance and tracking systems in autonomous probes [3], [4]. Today, these same methods are used in, e.g., autonomous vehicles to scan, detect, and model surrounding objects to prevent colisions [5], [6].

To track a target, we need to use some kind of sensor to collect information from the environment. The sensor can be of any kind [1]: a camera, lidar, radar, etc. If all we are interested in is the current position of the target, then a simple radar could be enough complete the task. This way, however, the information received would not be particularly precise as radar measurements can be affected by interferences from the environment, i.e., the measurements are noisy. We would also not be able to predict the movement of a target very accurately, if at all. Most importantly, we would not be able to automatically

[1] In this thesis we will use radars in examples.

detect a target since we would have no way of telling if a bright spot on an operator's screen is a reflection from a cloud, a mountain, or an airplane. For this reason, we need a more sophisticated method to process the incoming information. The method should be able to detect appearing and disappearing targets and handle uncertainty regarding the accuracy of radar measurements while also being able to predict future movement based on already acquired data. To achieve the last requirement, the method will have to infer properties or parameters of the target that are not directly observable, e.g., the velocity, acceleration, turn-rate, etc.

The Kalman filter proposed in [7] is one of such estimators. First proposed in 1960 by Rudolf E. Kalmán, it is still one of the most widely used estimators to this day. Its usage is not limited to target tracking only. To this day, it is being used in, e.g., predicting the state of a battery [8], [9] in electronic devices such as mobile phones or even electric vehicles. In its basic form, the Kalman filter applies only to linear models. For extended usage on non-linear problems, the Unscented Kalman and Extended Kalman filters [10] were proposed. In contrast with the linear Kalman filter, these variants are not optimal, i.e., covariances of predicted values are not minimal; moreover, in the case of the Extended Kalman filter, the filter may diverge completely. In target tracking, the Kalman filter can track only a single target. We do not allow any false alarm (false measurements) to appear in the observed region. Lastly, the filter cannot handle appearing or disappearing targets.

Even though the Kalman filter can be modified to handle multiple measurements by, e.g., choosing the nearest measurement to target the predicted location (NN-KF [11]), this approach is not ideal, as the filter can quickly diverge from the actual location if the nearest measurement is false. To address this issue more robustly, the probability data association (PDA) filter [12] has been proposed. The PDA filter computes an association probability for every measurement. Based on these probabilities, a weight is assigned to every measurement. This results in a weighted mixture of probability densities of possible locations of the target. This mixture is then combined using a moment-matching method, resulting in a final estimate of the target's location.

By solving the issue of processing multiple measurements at a time, the leap towards multiple target tracking filters becomes straightforward. The PDA filter is already suited for tracking multiple targets in our surveillance region. The simplest approach is to run the PDA filter for every target in the region independently. If these targets do not interact with each other, i.e., do not approach each other, the PDA filter will not be affected as distant measurements will be assigned weights close to zero, thus not interfering with its predictions. However, if these targets were to cross trajectories or move side-by-side, the association weights would suddenly increase to a non-insignificant value, resulting in fusing the predicted position of both targets to appear between them. This phenomenon is also called track coalescence. In such a situation, the PDA filter's accuracy is reduced. It will not be able to handle

the possible separation of said targets. A modification to the PDA filter was presented to minimize the probability of track coalescence occurring. The PDA filters cannot run isolated from each other; the association probabilities have to account for the possibility of a measurement originating from a different target, i.e., the association probabilities have to be computed jointly. The joint probability data association (JPDA) filter [12] works with the premise that only a single measurement can originate from a target at any given time. If two targets get sufficiently close to each other, the JPDA filter computes all possible association hypotheses with their respective probabilities. By marginalizing over the hypotheses, the JPDA filter computes the association weights of each measurement for each target, considering the possibility of a measurement originating from a different target. Once these weights are computed, the filter continues to form a probability density mixture for each target and merge them as in the PDA variant.

Being able to track multiple targets, the JPDA filter is still unable to detect and handle the appearing and disappearing targets. For this purpose, another modification of the joint integrated data association (JIPDA) filter [13] and IPDA [14] for the PDA filter variant was introduced. The integrated variant assigns a Bernoulli probability of existence to every target. With every timestep, the target is expected to survive with an arbitrary probability. If a measurement is associated with the given target, its probability is updated (increased) proportionally to the likelihood of the associated measurement. A target is then declared nonexisting if its existence probability falls below a specific value. On the other hand, in the JIPDA and IPDA approach, every measurement that is not associated with any target at an arbitrary timestep is declared a new target with its existence probability set reasonably close to the existence threshold. This ensures that measurements falsely declared a target will be removed in subsequent timesteps.

One could argue that the JPDA and PDA filters – while offering good tracking accuracy – approach the problem of measurement association as hungry algorithms. After every timestep, we merge all possible associations into a single ”pseudo” association, thus choosing the local optima. Instead, we could keep track of every association hypothesis separately, and with every incoming measurement set, we could incorporate them into every hypothesis separately. This way, we could filter out all wrong association hypotheses as they would become less probable. This approach is known as the Multiple hypothesis tracker (MHT). The most basic form proposed by Reid [15]. While the mathematics is similar to JPDA, the MHT filter (the MHT filter computes multi-scan associaton hypotheses) creates child hypotheses with every measurement scan. This results in an exponential increase in the number of tracked hypotheses. To keep the computational performance at reasonable levels, very aggressive pruning methods must be implemented. One of the issues of MTH filters is their implementation complexity, as multiple sophisticated techniques, search techniques, and data structures have to be used to keep the computational

complexity reasonable [16].

JPDA, PDA, and subsequently, JIPDA and IPDA filters are examples of the association-based paradigm of the target tracking topic. These filters offer good tracking performance in an environment with a low rate of clutter measurements while also keeping the computational complexity is relatively low.

Another fundamentally different paradigm is the random finite set (RFS) approach, which has gained traction in recent decades. One of the simpler RFS-based filters is the probability density hypothesis (PHD) filter [17] and its Gaussian variant GM-PHD [18]. The PHD filter approximates a Bayesian filter as it propagates only the first moment of the first-order multi-object density. In the point process theory, the first moment is usually called the intensity, the probability density function in target tracking. We expect the PHD to follow the multi-object Poisson distribution approximately to derive a closed-form solution. The reasonably straightforward implementation and low computational complexity are some of its advantages. The computational complexity stays low as it only approximates the multi-object density using the PHD and because it does not need to compute association probabilities. The main drawback of the filter is its unstable prediction of several targets in the region, which is caused by the approximation of the mixture and a lack of a rigorous way to provide track history for targets. To address these issues, the cardinality PHD (CPHD) filter [19], [20] – which also propagates the second moment of the density – and trajectory CPHD (TCPHD) filter [21] have been developed.

Recent developments in smart devices equipped with sensors and ever-increasing computing power has driven development in the field of sensor networks and smart grids. These networks allow us to collect data using different instruments, e.g., radar with lidar and camera, that can be placed in different locations and process it in a distributed and collaborative manner, e.g., autonomous vehicles and radar arrays.

The main approaches to collaborative filtering can be divided into several types: centralized architectures with a fusion center, incremental strategy with the Hamiltonian cycle, consensus-based, diffusion-based. In the centralized approach, each node with a sensor contains a local instance of a target-tracking agent. Data from all agents is then sent to a single central node, where it is processed, and the final estimate is computed [22]. An example of a fusion center-based Kalman filter can be found in [23] or in [24] for a PDA-based filter. The incremental strategy with the Hamiltonian cycle does not offer high robustness, as all nodes must form a cycle, thus rendering the network vulnerable to link failures. Additionally, establishing the Hamiltonian cycle is an NP-complete problem, which can severely affect performance in case of a link failure, as the cycle has to be computed anew. Lastly, the network limits the node-to-node communication to only a single neighbor, hindering the propagation of information and limiting the potential performance of the whole

network. The consensus and diffusion strategies are so-called fully distributed strategies. They do not impose restrictions on the network itself. In the consensus-based strategy, every node tries to iteratively find a combination of all its neighbor estimates [25]. This iteration lasts until a consensus among all neighbors is found. Due to this nature, real-time tracking is problematic [26]. The information fusion in the diffusion strategy happens in a single step. Every node receives information from its neighbors and incorporates it immediately into its estimate. Therefore, real-time tracking is not as problematic as in the consensus strategy.

While RFS-based filters outperform association-based ones, their distributed variants exist, e.g., [27] [28]. Their high computational complexity and memory demand require more powerful sensor devices to handle the computations in real time. One could argue that simpler association-based filters collaborating together could significantly improve the filtering performance while keeping the computational complexity and memory demand significantly lower compared to their RFS-based alternatives.

Chapter 1

Prerequisites

To understand the basis on which tracking algorithms are built, we need to introduce several mathematical concepts from the probability theory and statistical processing first.

1.1 Bayesian theorem

Bayesian theorem is the most fundamental theorem that allows us to determine the probability of a hypothesis given additional information (i.e., observations).

▶ **Theorem 1.1.** *Assume a real random variable X and its realization x and parameter θ, the following is true:*

$$\pi(\theta|X = x) = \pi(\theta|x) = \frac{f(x|\theta)\pi(\theta)}{f(x)}, \quad f(x) > 0, \tag{1.1}$$

where

- $f(x)$ *is marginal density of observations*
- $\pi(\theta|x)$ *is the posterior density of* θ
- $\pi(\theta)$ *is prior density of* θ
- $f(x|\theta)$ *is the likelihood of observations*

Since $f(x)$ is normalization factor, we can simply Equation (1.1) into:

$$\pi(\theta|x) \propto f(x|\theta)\pi(\theta), \tag{1.2}$$

where $\propto$ stands for proportionality, i.e., equality up to the normalizing factor.

For example, in linear regression, X is the observed (modeled) variable, θ are estimated regression coefficients, and $f(x|\theta)$ is a Gaussian distribution. In

state-space modeling, X is the observed (modeled) variable, θ are estimated system states, and $f(x|\theta)$ is a corresponding distribution. In GPS-based localization of a static target, X is the raw location measurement, θ is the true location to be estimated, and $f(x|\theta)$ is a two- or three-dimensional Gaussian distribution.

▶ Note 1.2. "For convenience, we will not distinguish between random variables and their realizations and use lowercase letters universally. The particular case is clear from the context.

1.2 Prior, posterior distributions and conjugate prior

Adhering to the Bayesian sequential framework, we aim to estimate parameter θ using its prior distribution $\pi(\theta)$, representing accumulated information from all previous observations. Using Equation (1.2), we get posterior distribution $\pi(\theta|x)$ representing updated information about the parameter.

Unfortunately, one of the main problems of the Bayesian approach is the rare existence of analytically tractable or computationally low-cost solutions to the posterior distribution[26]. This is where the concepts of conjugate prior and exponential family of distributions help us find an analytical solution of the posterior.

▶ **Definiton 1.3** (Exponential family of distributions)**.** *A family $\{F_\theta\}$ of distributions of a random variable x parametrized by a scalar or multivariate parameter θ is said to form an exponential family if the probability density function can be written in the form*

$$f(x|\theta) = h(x)g(\theta)\exp\{\eta(\theta)^\intercal T(x)\}, \tag{1.3}$$

where $h(x)$ is the base measure, $T(x)$ is a sufficient statistic, $g(\theta)$ is a known normalizing function and $\nu(\theta)$ is the natural parameter. If $\eta(\theta) = \theta$, then the family is called canonical [26]

▶ **Definiton 1.4** (Conjugate prior)**.** *Let $f(x|\theta)$ be a member of exponential family of distributions. Prior distribution $\pi(\theta)$ with hyperparameters ξ, ν is conjugate to it if its probability density has form,*

$$\pi(\theta) = q(\xi, \nu)g(\theta)^\nu \exp\{\eta(\theta)^\intercal \xi\}, \tag{1.4}$$

where ξ is a hyperparameter of the same size as $T(x)$, $\nu \in \mathbb{R}^+$ is scalar hyperparameter, $q(\xi, \nu)$ is a known function and $g(\theta)$ is the same function as in the exponential family distribution.

By choosing our model $f(x|\theta)$ from the exponential family of distributions and its appropriate conjugate prior $\pi(\theta)$, we ensure that the resulting posterior distribution also belongs to the exponential family and that the Bayesian update (1.2) has an analytical solution.

It is easy to show that the Bayesian update under conjugancy reduces to a simple update of hyperparameters ξ and ν [26] as:

$$\begin{aligned} \xi_k &= \xi_{k-1} + T(x_k), \\ \nu_k &= \nu_{k-1} + 1. \end{aligned} \tag{1.5}$$

1.3 Kullback-Leilbler divergence

Let $f(x)$ and $g(x)$ be probability density functions of a random variable x. Their Kullback-Leilbler divergence is then defined as

$$\mathcal{D}(f||g) = \mathbb{E}_{f(x)}\left[\log \frac{f(x)}{g(x)}\right] = \int f(x) \log \frac{f(x)}{g(x)} \mathrm{d}x.$$

The KL-divergence allows us to compare two probability density functions. It does not satisfy Euclid's triangle inequality, nor is it symmetric. Despite this, it is still widely used as a "similarity measure" in the Bayesian approach to probability.

Chapter 2

Single target tracking

In Chapter 1, model parameters represented by θ were considered static. This is, however, not the case in real-world scenarios where we are dealing with a dynamic stochastic system. Our main goal is to estimate the system's state with each incoming set. The state itself is usually not directly observable; therefore, we have to estimate it using a sequence of noisy measurements. For example, when we are tracking a plane in the sky, the state model may include x, y, z coordinates, velocity and acceleration in each direction. Yet the radar or other sensor can only provide the plane's position, and the rest of the parameters must be inferred. This process of state estimation is commonly referred to as filtering.

In this chapter, we will gradually lay the foundations of state representation, how we can connect received information (measurements) with the unknown state, which itself is not directly observable, to improve our knowledge of its parameters. Next, we will describe and derive one of the most fundamental filter algorithms, the Kalman filter. We will introduce the concept of clutter and its modeling and improve upon the Kalman filter to derive the PDA filter, which is built to work better in cluttered environments.

2.1 State-space model

Before tracking the model's state with incoming measurements, we must make two informed decisions. First, we need to know or estimate the model's state at a starting point in time k_0. Second, we must decide which evolution model best suits our tracking scenario.

Because these evolutionary models describe only the unobservable (hidden) model, they are also accompanied by the measurement model, which describes how the incoming measurements relate to the estimated model. The evolutionary and measurement models together form the state-space model. The most commonly used state-space models are the constant velocity model (CVM)

and constant acceleration model (CAM), which are shown in Examples 2.1 and 2.2.

▶ **Example 2.1** (CVM)**.** The most common for target tracking is called the constant velocity model [16]. The state vector contains only the position and velocity of the target. For a two-dimensional scenario, the state vector has the following form:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}, \tag{2.1}$$

$$\mathbf{x}_k = \begin{bmatrix} 1 & 0 & \Delta k & 0 \\ 0 & 1 & 0 & \Delta k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + v_k, \tag{2.2}$$

$$\mathbf{z}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{x}_k + w_k, \tag{2.3}$$

where $v_k \sim \mathcal{N}(0, Q_k)$ and w_k are the process and measurement noise variable, respectively.

▶ **Example 2.2** (CAM)**.**

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ v_x \\ v_y \\ a_x \\ a_y \end{bmatrix}, \tag{2.4}$$

$$\mathbf{x}_k = \begin{bmatrix} 1 & 0 & \Delta k & 0 & \frac{1}{2}\Delta k^2 & 0 \\ 0 & 1 & 0 & \Delta k & 0 & \frac{1}{2}\Delta k^2 \\ 0 & 0 & 1 & 0 & \Delta k & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta k \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + v_k, \tag{2.5}$$

$$\mathbf{z}_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}_k + w_k, \tag{2.6}$$

where $v_k \sim \mathcal{N}(0, Q_k)$ and w_k are defined as above.

2.1.1 Process model

The hidden process model can be described using the following equation

$$\mathbf{x}_k = F\mathbf{x}_{k-1} + v_k, \tag{2.7}$$

where $\mathbf{x}$ is the state vector times k and $k-1$, and F is a known transition matrix. Since no mathematical model is perfect, there will always be a part of real-world dynamics that cannot be described and modeled precisely. This indescribable part is, therefore, considered a noise of the model itself and is included in the noise term v_k.

To be able to derive a closed-form solution for the Kalman filter later in this chapter using concepts described in Chapter 1, we also have to assume that v_k is independent, centered at zero and Gaussian, i.e., white noise:

$$v_k \sim \mathcal{N}(0, \mathcal{Q}) \tag{2.8}$$

Since v_k is a Gaussian variable and x_k is a random variable, using a linear transformation, we get:

$$x_k \sim \mathcal{N}(Fx_{k-1}, \mathcal{Q}), \tag{2.9}$$

with its distribution represented by a probability density function $p(x_k|x_{k-1})$.

2.1.2 Measurement model

Having described the model itself, we now have to describe the relation of incoming measurements to the model's state. To do so, we have to introduce a new measured variable $\mathbf{z}_k$, defined as

$$\mathbf{z}_k = H\mathbf{x}_k + w_k, \tag{2.10}$$

where $\mathbf{x}_k$ is a state vector, H is a matrix of compatible shape, also known as the measurement matrix, w_k is the noise the sensor induces. As in the process model, we assume the measurement noise is Gaussian and centered at 0.

$$w_k \sim \mathcal{N}(0, \mathcal{R}) \tag{2.11}$$

Once again, using a linear transformation as in 2.9 we get:

$$\mathbf{z}_k \sim \mathcal{N}(H\mathbf{x}_k, \mathcal{R}), \tag{2.12}$$

with probability distribution function $p(\mathbf{z}_k|\mathbf{x}_k)$.

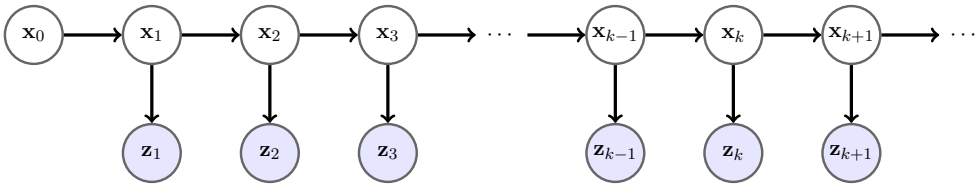


Figure 2.1 Graphical visualization of the Bayesian model estimation

2.1.3 Markov property

Lastly, we assume that both process and measurement models with their respective densities as $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and $p(\mathbf{z}_k|\mathbf{x}_k)$ comply with the Markov property. The Markov property states that the state at time k is directly dependent only on the previous state at time $k-1$:

$$p(\mathbf{x}_k|\mathbf{x}_1,\ldots,\mathbf{x}_{k-2},\mathbf{x}_{k-1},\mathbf{z}_1,\ldots,\mathbf{z}_{k-2},\mathbf{z}_{k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1}), \tag{2.13}$$

$$p(\mathbf{z}_k|\mathbf{x}_1,\ldots,\mathbf{x}_{k-2},\mathbf{x}_{k-1},\mathbf{x}_k,\mathbf{z}_1,\ldots,\mathbf{z}_{k-2},\mathbf{z}_{k-1}) = p(\mathbf{z}_k|\mathbf{x}_k), \tag{2.14}$$

i.e., we assume that the information known now or in the past does not provide information about the future behavior of the target. Due to this assumption, the hidden process model is also called the hidden Markovian model.

2.2 Kalman filter

Kalman filter is one of the most basic filtering algorithms. Being used first in the Apollo navigation program, it continues to be one of the most fundamental sequential estimation and filtering algorithms in 3D modeling, time-series analysis, electronics, and many other fields. Its use is not limited to state estimation only, but it has proven to be useful in signal smoothing, e.g., the remaining battery time indication. The Kalman filter in its basic form, which will be described in this chapter, applies to linear problems only; however, non-linear formulations such as extended Kalman filter or unscented Kalman filter do exist.

2.2.1 Derivation

Like most filtering algorithms, the Kalman filter estimates the process model using feedback control. First, the filter estimates the process state at time k, and once a measurement from a sensor arrives, it corrects its predicted state in a Bayesian fashion. Note that if a measurement at timestep k and/or any subsequent timesteps won't arrive, the filter will keep predicting the state based on measurements that have arrived until that point.

Given the process model and measurements model from the previous section

$$\mathbf{x}_k = F\mathbf{x}_{k-1} + v_k, \qquad v_k \sim \mathcal{N}(0, \mathcal{Q}) \tag{2.15}$$
$$\mathbf{z}_k = H\mathbf{x}_k + w_k, \qquad w_k \sim \mathcal{N}(0, \mathcal{R}) \tag{2.16}$$
$$x_0 = \mathcal{N}(\hat{\mathbf{x}}_0, P_0), \tag{2.17}$$

process and measurement noise represented as white noise and random variables x_k and z_k as:

$$\mathbf{x}_k \sim \mathcal{N}(F\mathbf{x}_{k-1}, \mathcal{Q}), \qquad \text{with probability density } p(\mathbf{x}_k|\mathbf{x}_{k-1}) \tag{2.18}$$
$$\mathbf{z}_k \sim \mathcal{N}(H\mathbf{x}_k, \mathcal{R}), \qquad \text{with probability density } p(\mathbf{z}_k|\mathbf{x}_k). \tag{2.19}$$

We can derive analytical solutions to the prediction and update steps.

2.2.2 Prediction step

The prediction step, also known as the Chapmap-Kolmogorov equation, updated the posterior distribution with the evolution model:

$$p(\mathbf{x}_k|\mathbf{z}_{0:k-1}) = \int p(\mathbf{x}_k, \mathbf{z}_{k-1}|\mathbf{z}_{0:k-1}) d\mathbf{x}_{k-1} \tag{2.20}$$
$$= \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{z}_{k-1}) d\mathbf{x}_{k-1}. \tag{2.21}$$

Multiplying two Gaussian distributions yields a single multivariate Gaussian distribution. By marginalizing the multivariate Gaussian over x_{k-1} we get a prior Gaussian distribution $\mathcal{N}(\mathbf{x}_{k|k-1}, P_{k|k-1})$ with parameters

$$\mathbf{x}_{k|k-1} = F\mathbf{x}_k + \mathcal{Q}, \tag{2.22}$$
$$P_{k|k-1} = HP_kH^T + \mathcal{R}. \tag{2.23}$$

2.2.3 Update step

The filter uses the Bayesian theorem to correct the estimate based on the new measurement z_k in the update step.

$$p(\mathbf{x}_k|\mathbf{z}_{0:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}_{0:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{0:k-1})} \tag{2.24}$$
$$\propto p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}_{0:k-1}). \tag{2.25}$$

From Section 1.2, we know that if the prior is conjugate, the Bayesian update can be reduced to a sum of hyperparameters and sufficient statistics.

We must rewrite the model and prior distributions in exponential form to

get the desired result. For the model distribution, we get

$$\begin{aligned} p(\mathbf{z}_k|\mathbf{x}_k) &\propto \exp\left\{-\frac{1}{2}(\mathbf{z}_k - H\mathbf{x}_k)^\intercal \mathcal{R}^{-1}(\mathbf{z}_k - H\mathbf{x}_k)\right\} \\ &= \exp\left\{Tr\Bigg(\underbrace{-\frac{1}{2}\begin{bmatrix}-1\\ \mathbf{x}_k\end{bmatrix}\begin{bmatrix}-1\\ \mathbf{x}_k\end{bmatrix}^\intercal \begin{bmatrix}\mathbf{z}_k^\intercal \\ H^\intercal\end{bmatrix}}_{\eta} \underbrace{\mathcal{R}^{-1}\begin{bmatrix}\mathbf{z}_k^\intercal \\ H^\intercal\end{bmatrix}^\intercal}_{T(\mathbf{z}_k)}\Bigg)\right\}, \end{aligned} \tag{2.26}$$

and for the prior distribution

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{z}_{0:k-1}) &\propto \exp\left\{-\frac{1}{2}(\mathbf{x}_k - \mathbf{x}_{k|k-1})^\intercal P_{k|k-1}^{-1}(\mathbf{x}_k - \mathbf{x}_{k|k-1}\right\} \\ &= \exp\left\{Tr\Bigg(\underbrace{-\frac{1}{2}\begin{bmatrix}-1\\ \mathbf{x}_k\end{bmatrix}\begin{bmatrix}-1\\ \mathbf{x}_k\end{bmatrix}^\intercal}_{\eta} \underbrace{\begin{bmatrix}\mathbf{x}_{k|k-1}^\intercal \\ I\end{bmatrix} P_{k|k-1}^{-1}\begin{bmatrix}\mathbf{x}_{k|k-1}^\intercal \\ I\end{bmatrix}^\intercal}_{\xi_k}\Bigg)\right\}, \end{aligned} \tag{2.27}$$

where I is the identity matrix.

Now that we have derived forms of the sufficient statistic and hyperparameter ξ_k, we can write the Bayesian update as

$$\begin{aligned} \xi_k &= \xi_{k-1} + T(\mathbf{z}_k) \\ &= \begin{bmatrix} \mathbf{x}_{k|k-1}^\intercal P_{k|k-1}^{-1}\mathbf{x}_{k|k-1} + \mathbf{z}_k^\intercal \mathcal{R}^{-1}\mathbf{z}_k, & \mathbf{x}_{k|k-1}^\intercal P_{k|k-1}^{-1} + \mathbf{z}_k^\intercal \mathcal{R}^{-1} H \\ P_{k|k-1}^{-1}\mathbf{x}_{k|k-1}^{-1} + H^\intercal \mathcal{R}^{-1}\mathbf{z}_k, & P_{k|k-1}^{-1} + H^\intercal \mathcal{R}^{-1} H \end{bmatrix}. \end{aligned} \tag{2.28}$$

Parameters of the posterior distribution can be derived as

$$\begin{aligned} P_k &= (\xi_{k;[2,2]})^{-1} \\ &= [P_{k|k-1}^{-1} + H^\intercal \mathcal{R}^{-1} H]^{-1} \\ &= (I - W_k H) P_{k|k-1}, \end{aligned} \tag{2.29}$$

$$\begin{aligned} \mathbf{x}_k &= (\xi_{k;[2,2]}^{-1})\xi_{k;[2,1]} \\ &= P_k[P_{k|k-1}^{-1}\mathbf{x}_{k|k-1}^\intercal + H^\intercal \mathcal{R}^{-1}\mathbf{z}_k] \\ &= \mathbf{x}_{k|k-1} + P_k H^\intercal \mathcal{R}^{-1}(\mathbf{z}_k - H\mathbf{x}_{k|k-1}). \end{aligned} \tag{2.30}$$

where

$$W_k = P_{k|k-1}H^\intercal (R + H P_{k|k-1} H^\intercal)^{-1} \tag{2.31}$$

is called the Kalman gain. With growing Kalman gain, newer measurements are considered more relevant, making the filter more sensitive but less able to filter out incoming noise. Another relevant way of thinking about the Kalman gain is that the actual measurement z_k is trusted more as the covariance $\mathcal{R}$

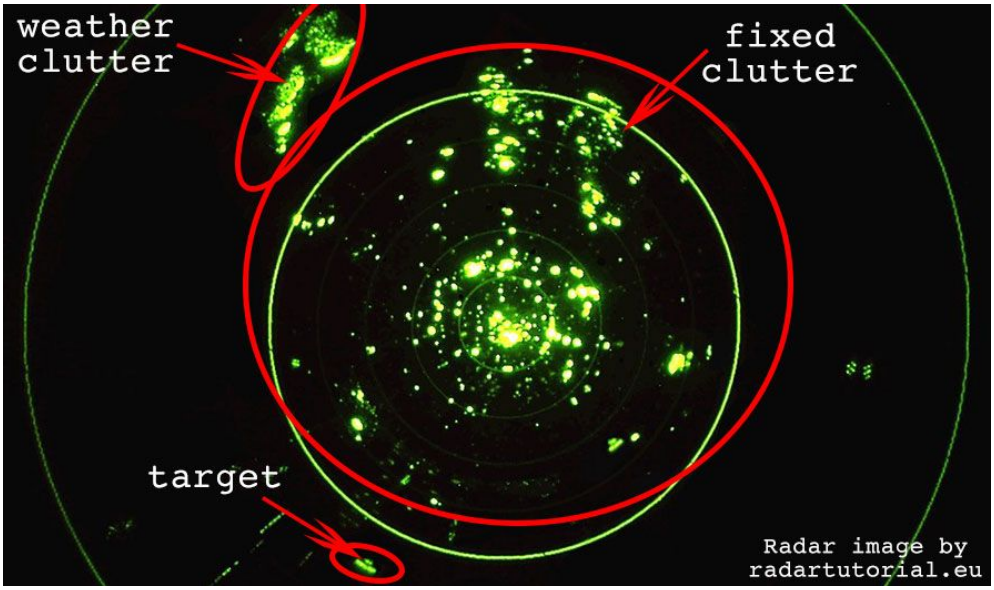


Figure 2.2 Annotated radar clutter (Source: www.radartutorial.eu)

approaches zero, while the predicted measurement $H\mathbf{x}_{k|k-1}$ is trusted less [29].

2.3 Clutter

In many target-tracking applications, measurements may not necessarily originate from a currently tracked target. This is especially true in multiple target tracking scenarios, where targets appear and disappear as time goes on. Additionally, the tracking algorithm should be able to work with and potentially filter out so-called false alarms. These measurements do not necessarily originate from a target; instead, they result from the imperfection of the radar itself. This phenomenon can be even amplified if we need to increase the radar's sensitivity, e.g., if we are tracking objects that are hard to detect. These measurements with unknown origin are commonly referred to as clutter.

To incorporate our beliefs and estimates into the tracking algorithm, we must first choose a model for the clutter behavior. Now, let us focus on where we are using radar as a sensor. Measurements that are usually fed to the tracking algorithm are extracted from a radar image. The image consists of many resolution cells, each having a Boolean detector. The detector is responsible for reporting if the target resides in its cell. Since no sensor is perfect, we expect some cells to misreport a residing target. We define a false alarm rate P_{FA} to quantify this phenomenon. Typical value of the P_{FA} is between 10^{-6} and 10^{-2} [16]. We also assume that these false detections are are i.i.d. in different cells. With these assumptions, the total number of false

alarms (ϕ) is distributed according to the binomial distribution

$$\mu(\phi) = \binom{N}{\phi} P_{FA}^{\phi}(1 - P_{FA})^{N-\phi}, \tag{2.32}$$

where N is the total number of resolution cells. We also expect the N to be a large number, and since P_{FA} is small, we can approximate the binomial distribution using Poisson distribution. Defining $\Lambda = NP_{FA}$, we can express the clutter as

$$\mu(\phi) = e^{-\Lambda}\frac{\Lambda^{\phi}}{\phi!} \tag{2.33}$$

Moving a step forward, we take into account our region of interest $\mathfrak{R}$ and its volume V. We can then define $\lambda = \frac{\Lambda}{V}$ and by substituting for Λ in Equation (2.33) we obtain

$$\mu(\phi) = e^{-(\lambda V)}\frac{(\lambda V)^{\phi}}{\phi!}. \tag{2.34}$$

This allows us to model the clutter cardinality without necessarily knowing P_{FA} and N. Instead, we need to know the region's volume of interest V and estimate λ. Equation (2.34) also relates closer to the standard way of modeling clutter, the homogenous Poisson point process (HPPP) with intensity λ.

Lastly, we must model the spatial distribution of clutter measurement along region $\mathfrak{R}$. To do so, we can use a property of HPPP, which states that the position of events will be uniformly distributed [30]. The whole process of modeling clutter is presented in pseudocode 1

Algorithm 1 Generating clutter using the HPPP

Require: $0 < \lambda < 1, V > 0$

$Z_k \leftarrow \emptyset$
$m_k \leftarrow Po(\lambda V)$
for $j \leftarrow 1, m_k$ **do**
    $z_k^j \leftarrow Unif(V)$
    $Z_k \leftarrow Z_k + z_k^j$
end for

2.4 Misdetections

Another complication in the target tracking problem is that the target may or may not be detected at any timestep with a certain probability. We assume that the detection events (i.e., if the target is detected) are independent at

times k, l when $k \neq l$. Given this assumption, a simple detection model using a Bernoulli random variable

$$P(\delta) = \begin{cases} P_D & if\ \delta = 1, \\ 1 - P_D & if\ \delta = 0. \end{cases} \tag{2.35}$$

However, this creates another complication in choosing the correct value of the underlying parameter P_D. Typical values can range from 0.5 for sonar tracking applications to 0.95 for radar tracking of airplanes. The clutter intensity λ and detection probability P_D are not independent. We can lower the detection threshold on a sensor to increase detection probability. However, this also increases the rate of false alarms. Choosing the best value is a matter of engineering judgment [16].

2.5 Probabilistic Data Association filter

Even though the Kalman filter can handle misdetections of a target, it falls short when false measurements (clutter) are introduced to the measurement process, since it must associate the only measurement with the target.

The simplest heuristic solution to this problem would be the nearest neighbor Kalman filter (NN-KF), which selects the closest measurement, based on an arbitrary distance metric, to the predicted position as the correct one. However, this approach is viable only when false measurements are low. We can also view it as choosing local optima in every timestep, which leads to a high chance of the filter being led astray from the trajectory of the actual target.

In general, two other options exist for selecting optimal associations. The first option is to select measurement to target the association in a way that maximizes the posterior distribution probability (i.e., maximum posterior method or MAP), the second option, which the PDAF is based on, is minimizing the mean square error over all possible associations [12].

2.5.1 Model assumptions

Before we describe the mathematical principles behind the PDA filter, we need to address several assumptions about the process model and measurement model that need to be made [16], [12]:

1. If multiple measurements fall inside the validation region of a target, then at most one of the measurements is considered to come from the target.
2. The remaining measurements are considered false alarms or clutter and are modeled as i.i.d. with uniform spatial distribution. The number of false targets obeys the Poisson process with spatial density λ.
3. The prior density of $\mathbf{x}_{k-1}$ is given as $p_{k|k-1}(\mathbf{x}_{k-1})$.

4. The state vector evolves according to the Markov process model $f(x_k|x_{k-1})$.
5. If a target-originated measurement exists, it is related to x_k according to the likelihood of $f(z_k|x_k)$.
6. Target detections occur independently over time with probability P_D.
7. The track has already been initialized
8. At time step $k-1$ only one target exists in the surveillance region with state x_{k-1}.

Additionally, we assume the prior and posterior models to be linear and Gaussian with their pdfs

$$\begin{aligned} p_{k|k-1}(x_k) &= \mathcal{N}(x_k; \hat{x}_{k|k-1}, P_{k|k-1}), \\ f(x_k|\mathbf{x}_{k|k-1}) &= \mathcal{N}(x_k, Fx_{k-1}, \mathcal{Q}), \\ f(z_k|x_k) &= \mathcal{N}(z_k, Hx_k, \mathcal{R}), \end{aligned} \tag{2.36}$$

thus ensuring that the prediction and update steps will have analytical solutions. Lastly, we define m_k as a number of measurements a time k and a set of all measurements at the time k as $Z_k = \{z_k^1, z_k^2, \ldots, z_k^{m_k}\}$.

2.5.2 Posterior density as a mixture

According to [16], the posterior density can be written as

$$p_{k|k}(x_k) = \sum_{i=0}^{m_k} p(x_k|a_k = i, Z_{0:k}) Pr\{a_k = i|Z_{0:k}\}, \tag{2.37}$$

where $a_k = i,\ i \in \{0, 1, \ldots, m_k\}$ means that the measurement i originates from the target, with $a_k = 0$ meaning no measurement originates from the target, and $Z_{0:k}$ represents the collection of measurement sets received at timestep k.

For notational convenience, we will further confuse the random variable a_k and its realization, thus simplifying the previous equation to

$$p_{k|k}(x_k) = \sum_{a_k} p(x_k|a_k, Z_{0:k}) Pr\{a_k|Z_{0:k}\}. \tag{2.38}$$

Equation (2.38) shows that the PDA filter calculated probabilities of all possible realizations of the variable a_k and creates a weighted posterior mixture. The PDA filter aims to reduce this mixture into a single posterior distribution.

2.5.3 PDA posterior

As is in the case with the Kalman filter, the prior density of the PDA filter is given by the Chapman-Kolmogorov equation

$$p_{k|k-1}(x_k) = \int f(x_k|x_{k-1})p_k(x_{k-1})dx_{k-1}. \tag{2.39}$$

The form posterior density in the PDA filter is dependent on the detection event.

$$p_{k|k}(x_k|a_k) \propto \begin{cases} p_{k|k-1}(x_k) & if\ a_k = 0, \\ f(z_k^{a_k}|x_k)p_{k|k-1}(x_k) & if\ a_k > 0. \end{cases} \tag{2.40}$$

The notation $z_k^{a_k}$ implies that the number a_k of the m_k measurements is used.

If at timestep k, no measurement is assigned to the target. The prior distribution remains unchanged and continues to be used as posterior distribution in the subsequent step. If a measurement from Z_k is assigned to the target, then Equation (2.40) leads to an update of the prior distribution in a Bayesian manner. Assuming a linear Gaussian model, the second line reduces to the Kalman filter update.

2.5.4 Validation gating

Until now, we have considered that every single measurement across the surveilled region could originate from a given target. While we measure how related a measurement is using likelihood (Assumption 5), limiting the considered region to a "reasonable" distance from the predicted target position would make practical sense. The likelihood of measurements falling outside this validation region would be near zero, thus making them irrelevant. According to [16] and [12], the region is usually constructed as an ellipsoid whose shape is given by the covariance matrix S_k and scaling parameter γ. The validation region would be then defined as

$$\mathfrak{L} = \{z : (z_k^i - \hat{z}_{k|k-1})S_k^{-1}(z_k^i - \hat{z}_{k|k-1}) < \gamma\}. \tag{2.41}$$

However, finding the optimal value of γ can be cumbersome. Let us, therefore define $\zeta_i = z_k^i - \hat{z}_{k|k-1}$ and $n = \dim \mathfrak{R}$. Since measurements are corrupted by a zero-mean noise with covariance matrix S, the ζ_i can be viewed as noise realization

$$\zeta_{i,k|k-1} \sim \mathcal{N}(0, S). \tag{2.42}$$

We aim to find a value of γ such that

$$\begin{aligned}(z_k^i - \hat{z}_{k|k-1})^\intercal S^{-1}(z_k^i - \hat{z}_{k|k-1}) &= \zeta_i^\intercal S^{-1}\zeta_i \\ &= \underbrace{\zeta_i^\intercal U^{-\intercal}}_{u^\intercal}\underbrace{U^{-1}\zeta_i}_{u} \\ &= u^\intercal u \\ &\leq \gamma.\end{aligned} \tag{2.43}$$

Looking closer at the expression $U^{-1}(z_i - \hat{z})$, we can see that it corresponds to a standardization of a random variable.

$$\frac{X-\mu}{\sigma} = \underbrace{\sqrt{\sigma^2}(x-\mu)}_{\downarrow} \sim \mathcal{N}(0,1), \tag{2.44}$$

$$U^{-1}(z_i - \hat{z}) = U^{-1}\zeta_i = u \sim \mathcal{N}(0, I), \tag{2.45}$$

This means that elements of u are i.i.d. and $\mathcal{N}(0,1)$ and $u^\intercal u$ is sum of squares of the elements u. The sum of squared $\mathcal{N}(0,1)$ variables has, by definition, χ^2_n distribution with degrees of freedom corresponding to the number of summands:

$$u^\intercal u = \sum_{i=1}^{n} u_i^2 \sim \chi^2_n. \tag{2.46}$$

This allows us to define the distance threshold γ using the gating probability $P_G \in (0,1]$ using the inverse of the χ^2 distribution function F,

$$\gamma = u^\intercal u = F^{-1}(P_G). \tag{2.47}$$

2.5.5 Data association

Assuming linear, Gaussian model and clutter density given by Poisson process, the association probability follows Equation [16]:

$$\beta_i^{a_k} = P\{a_k | Z_{0:k}\} \propto \begin{cases} \lambda(1 - P_D P_G) & if\ a_k = 0, \\ P_D \mathcal{N}(z_k^{a_k}; \hat{z}_{k|k-1}, S_k) & if\ a_k > 0, \end{cases} \tag{2.48}$$

where $\hat{z}_{k|k-1}$ and S_k are predicted measurement and its covariance at timestep k given by

$$\hat{z}_{k|k-1} = H\hat{x}_{k|k-1}, \tag{2.49}$$

$$S_k = HP_{k|k-1}H^\intercal + \mathcal{R}. \tag{2.50}$$

2.5.6 Mixture reduction

In the last step of the PDAF cycle, we need to reduce the Gaussian mixture into a single posterior distribution. The PDAF uses the moment matching method to determine the first and second moments of the final Gaussian distribution.[16]

First, let us define event-conditional innovation as

$$\nu_k^{a_k} = z_k^{a_k} - H\hat{x}_{k|k-1}, \tag{2.51}$$

and the event-conditional state estimate of $\hat{x}_k^{a_k}$ and its covariance $P_k^{a_k}$ as

$$\hat{x}_k^{a_k} = \begin{cases} \hat{x}_{k|k-1} & if\ a_k = 0, \\ \hat{x}_k + W_k\nu_k^{a_k} & if\ a_k > 0, \end{cases} \tag{2.52}$$

$$P_k^{a_k} = \begin{cases} P_{k|k-1} & if\ a_k = 0, \\ (I - \mathbf{W}_k H)P_{k|k-1} & if\ a_k > 0. \end{cases} \tag{2.53}$$

The expectation of the mixture is

$$\hat{x}_k = \beta_k^0\hat{x}_{k|k-1} + \mathbf{W}_k\nu_k, \tag{2.54}$$

where $\nu_k = \sum_{a_k>0}\beta_k^{a_k}\nu_k^{a_k}$ and $\mathbf{W}_k$ is the Kalman gain. Matching covariances of the mixture leads to

$$P_k = P_{k|k-1} - (1 - \beta_0^k)\mathbf{W}_k S_k \mathbf{W}_k + \tilde{P}_k, \tag{2.55}$$

where

$$\tilde{P}_k = \mathbf{W}_k \left[\sum_{a_k>0} \beta_k^{a_k}\nu_k^{a_k}(\nu_k^{a_k})^\intercal - \nu_k\nu_k^\intercal \right] \mathbf{W}_k^\intercal, \tag{2.56}$$

as the spread of innovations must be accounted for when merging multiple distributions. The final posterior density becomes

$$p(x_k|z_{0:k}) \approx \mathcal{N}(x_k; \hat{x}_k, P_k). \tag{2.57}$$

2.6 Track management and IPDA

The PDA filter tells us how to proceed with the filter update if multiple measurements fall inside the validation region. However, it relies on the assumption that the track has already been initialized and does not offer a solution to the initialization problem.

Since all target tracks are generally considered a priory uninitialized, we require that the tracking system itself should determine if a series of measurements originate from a target or not and prospectively initialize a new track, as well as determine when a target is no longer visible and terminate its track accordingly.

2.6.1 Heuristic approach

One of the most common and straightforward heuristic approaches is the so-called 2/2 & M/N rule. [31] Having received two consecutive measurements in proximity, determined, for example, by the maximal velocity of a target, we create a candidate track, which the PDAF handles. After N additional timesteps, if a measurement has fallen in the validation region in M steps, the track is declared as confirmed or terminated otherwise.

Confirmed tracks are no longer held subject to the same test; however, a confirmed track may disappear due to the target leaving the surveillance region, remaining hidden behind an obstacle, or the tracks failing to follow the correct measurement, etc. In this case, the same rule is also applicable to termination logic.

The simplicity of the M/N rule approach comes at the cost of the potentially significant delay in the track initialization and termination. For this reason, several different approaches have been developed to achieve faster decision-making than the M/N rule. One of the representatives of the existence-based approach is the integrated probability data association (IPDA), which is described in the following section.

2.6.2 IPDA

In contrast with the former approach, the IPDA filter does not necessarily seek to make a binary decision on whether the target exists. Instead, it aims to quantify and estimate the probability of the existence of the target. This gives us the freedom to either use the estimated existence probability to make the decision or to use the probability as it is and potentially display it with the target to the system user.

The existence-based approach has dominated target-tracking research since the 90s, mainly due to advances in the modern paradigm of multiple target tracking with existence probability expressed in random finite sets. The IPDA, proposed in 1994, was the first step in introducing the concept of track

existence probability into track-oriented filters (e.g., Kalman Filter, PDAF, JPDAF, etc.).

2.6.2.1 Existence probability

In the previous chapter, we built the PDA filter with several assumptions. Assumption 8 states that one and only one target exists in the surveillance region. This is, however, not the case with IPDA. Therefore, let us modify the assumption to conform to the new scenario:

Let the probability of the target existing at timestep $k-1$ be r_{k-1}, and $1-r_{k-1}$ be the probability that the target does not exist. If the target exists at timestep $k-1$, it continues to exist at timestep k with the probability P_s or ceases to exist with the probability $1-P_s$. If the target did not exist at timestep $k-1$, it also does not exist at timestep k [16].

The existence probability of the target can be modeled as the Markov process [32]. In the standard formulation, the Markov process allows targets to die but does not allow them to reemerge again.

2.6.2.2 Existence prediction

The prediction of target existence is again done using the Chapman-Kolmogorov equation. Since we are working with the Markov process, we can describe its behavior with the Markov chain transition matrix

$$\begin{bmatrix} r_{k|k-1} \\ 1-r_{k|k-1} \end{bmatrix} = \begin{bmatrix} P_s & 0 \\ 1-P_s & 1 \end{bmatrix} \begin{bmatrix} r_{k-1} \\ 1-r_{k-1} \end{bmatrix} = \begin{bmatrix} P_s r_{k-1} \\ 1-P_s r_{k-1} \end{bmatrix}, \tag{2.58}$$

where $r_{k|k-1}$ is the predicted existence probability.

2.6.2.3 Existence probability update

Let us now focus on the primary distinction of IPDA, the posterior existence probability. Even though IPDA uses the same association events as PDAF, we have to be more careful when defining the IPDA association probabilities conditionals as the existence of the tracked object is not guaranteed. Therefore, let us define two additional events: E (target exists) and D (target is detected). We define the IPDA association probabilities as

$$\beta_k^{a_k} = Pr\{a_k|E, Z_{0:k}\} = \frac{Pr\{a_k, E|Z_{0:k}\}}{Pr\{E|Z_{0:k}\}}. \tag{2.59}$$

Later in this chapter, we will show that the association probabilities of IPDA are the same as those of the PDA filter.

▶ **Theorem 2.3.** *Let $r_{k|k-1}$ be the predicted existence probability. Assuming that clutter behaves according to the Poisson point process and the model is*

linear-Gaussian, we can express the posterior existence probability as

$$r_k = \frac{\mathcal{L}_k r_{k|k-1}}{1 - (1 - \mathcal{L}_k) r_{k|k-1}}, \tag{2.60}$$

where $\mathcal{L}_k$ *corresponds to*

$$\mathcal{L}_k = 1 - P_D P_G + \frac{P_D P_G}{\lambda} \sum_{a_k}^{m_k} \mathcal{N}(z_k^{a_k}; \hat{z}_{k|k-1}, S_k). \tag{2.61}$$

Proof. Let us first focus on the different realizations of variable a_k. For IPDA, we can adapt Equation (2.48). For PDAF, the probability of detection was $P\{D\} = P_D$ and the probability of a measurement appearing in the validation region $P\{G\} = P_G$. For IPDA, we have to account for the probability of existence. Therefore, we get the following:

$$Pr\{D \cap G\} = Pr\{D \cap G|E\} Pr\{E\} = r_{k|k-1} P_D P_G, \tag{2.62}$$

when model pdfs, and clutter descriptions stay the same as in PDAF, we come to the conclusion that

$$Pr\{a_k|Z_{0:k}\} \propto \begin{cases} (1 - P_D P_G r_{k|k-1})\lambda & if \quad a_k = 0, \\ r_{k|k-1} P_D \mathcal{N}(z_k^{a_k}; \hat{z}_{k|k-1}, S_k) & if \quad a_k > 0. \end{cases} \tag{2.63}$$

From Equation (2.59), we know that association probabilities are proportional to the joint probabilities $Pr\{a_k|E, Z_{0:k}\}$. For the case where $a_k > 0$, i.e., the target is detected, the existence probability conditioned on a_k becomes unity, thus yielding

$$Pr\{a_k = i, E, Z_{0:k}\} = Pr\{a_k = i|Z_{0:k}\}. \tag{2.64}$$

In the case of $a_k = 0$, we have to make several manipulations have to be made to relate $Pr\{a_k = i, E, Z_{0:k}\}$ and $Pr\{a_k = i|Z_{0:k}\}$. First, we will need to express the existence probability if no measurement from the target was

detected

$$
\begin{aligned}
Pr\{E|a_k = 0\} &= Pr\{E|\neg D, Z_{0:k}\} && (2.65)\\
&= Pr\{E|\neg D, Z_{0:k-1}\} && (2.66)\\
&= \frac{Pr\{\neg|E\}Pr\{E|Z_{0:k-1}\}}{Pr\{\neg D|E\}Pr\{E|Z_{0:k-1}\} + Pr\{\neg D|\neg E\}Pr\{\neg E|Z_{0:k-1}\}} && (2.67)\\
&= \frac{(1 - P_D P_G) r_{k|k-1}}{(1 - P_D P_G) r_{k|k-1} + (1 - r_{k|k-1})} && (2.68)\\
&= \frac{r_{k|k-1}(1 - P_D P_G)}{1 - P_D P_G r_{k|k-1}}. && (2.69)
\end{aligned}
$$

We are now able to relate the joint probabilities as

$$
\begin{aligned}
Pr\{a_k = 0, E|Z_{0:k}\} &= Pr\{E|a_k = 0, Z_{0:k}\}Pr\{a_k = 0|Z_{0:k}\}\\
&= Pr\{E|a_k = 0, Z_{0:k-1}\}Pr\{a_k = 0|Z_{0:k}\}\\
&= \frac{Pr\{E|a_k = 0, Z_{0:k-1}\}}{Pr\{a_k = 0|Z_{0:k}\}} Pr\{a_k = 0|Z_{0:k}\}\\
&= \frac{r_{k|k-1}(1 - P_D P_G)}{1 - P_D P_G r_{k|k-1}} Pr\{a_k = |Z_{0:k}\}. && (2.70)
\end{aligned}
$$

To get the first equation, we utilized the definition of conditional probability. The second equation utilizes the fact that in the current case, all measurements are considered to be clutter; thus, the new measurement does not carry any new information about the existence of the target. In the third equation, we again used the definition of conditional probability. Finally, in the fourth equation, we can only express the fraction with $r_{k|k-1}$, P_D, and P_G. From this the association probabilities $\beta_k^{a_k}$ are given by

$$
\begin{aligned}
\beta_k^{a_k} &\propto Pr\{a_k, E|Z_{0:k}\}\\
&\propto \begin{cases} \frac{r_{k|k-1}(1-P_D P_G)}{1-P_D P_G r_{k|k-1}} Pr\{a_k|Z_{0:k}\} & if\ a_k = 0,\\ Pr\{a_k|Z_{0:k}\} & if\ a_k > 0, \end{cases}\\
&\propto \begin{cases} \frac{r_{k|k-1}(1-P_D P_G)}{1-P_D P_G r_{k|k-1}} (1 - P_D P_G r_{k|k-1})\lambda & if a_k = 0,\\ P_D r_{k|k-1} l^{a_k} & if\ a_k > 0, \end{cases}\\
&\propto \begin{cases} (1 - P_D P_G)\lambda & if\ a_k = 0,\\ P_D l^{a_k} & if\ a_k > 0, \end{cases} && (2.71)
\end{aligned}
$$

where $l^{a_k} = \mathcal{N}(z_k^{a_k}; \hat{z}_{k|k-1}, S_k)$.

Since the expression is identical, we have shown that IPDA and PDAF use the same probability associations. Lastly, we will derive the formula pro

posterior existence probability. From the total probability theorem, we can express the posterior probability as

$$\begin{aligned} Pr\{E|Z_{0:k}\} &= Pr\{E|a_k = 0, Z_{0:k}\}Pr\{a_k = 0|Z_{0:k}\} \\ &\quad + \sum_{i=1}^{m_k} Pr\{E|a_k = i, Z_{0:k}\}Pr\{a_k = i|Z_{0:k}\} \\ &= Pr\{E|a_k = 0, Z_{0:k}\}Pr\{a_k = 0|Z_{0:k}\} \\ &\quad + \sum_{i=1}^{m_k} Pr\{a_k = i|Z_{0:k}\}. \end{aligned} \tag{2.72}$$

Now using by incorporating Equations (2.63) and (2.70) into (2.72) we get

$$\begin{aligned} Pr\{E|Z_{0:k}\} &= \frac{\frac{r_{k|k-1}(1-P_DP_G)}{1-P_DP_Gr_{k|k-1}}(1 - P_DP_Gr_{k|k-1})\lambda + P_Dr_{k|k-1}\sum_{a_k=1}^{m_k} l^{a_k}}{1 - P_DP_Gr_{k|k-1} + r_{k|k-1}\frac{P_D}{\lambda}\sum_{i=1}^{m_k} l^{a_k}} \\ &= \frac{(1 - P_DP_G) + \frac{P_D}{\lambda}\sum_{i=1}^{m_k} l^{a^{a_k}}}{1 - P_DP_Gr_{k|k-1} + r_{k|k-1}\frac{P_D}{\lambda}\sum_{a_k=1}^{m_k} l^{a^{a_k}}} r_{k|k-1}, \end{aligned} \tag{2.73}$$

. The $\mathcal{L}_k$ is recognizable from the numerator. The denominator we have to modify further

$$\begin{aligned} &1 - P_DP_Gr_{k|k-1} + r_{k|k-1}\frac{P_D}{\lambda}\sum_{a_k=1}^{m_k} l^{a^{a_k}} r_{k|k-1} \\ &= 1 - (P_DP_G + \frac{P_DP_G}{\lambda}\sum_{a_k=1}^{m_k} l^{a^{a_k}})r_{k|k-1} \\ &= 1 - (1 - 1 + P_DP_G + \frac{P_DP_G}{\lambda}\sum_{a_k=1}^{m_k} l^{a^{a_k}})r_{k|k-1} \\ &= 1 - (1 - (1 - P_DP_G) + \frac{P_DP_G}{\lambda}\sum_{a_k=1}^{m_k} l^{a^{a_k}})r_{k|k-1} \\ &= 1 - (1 - \mathcal{L}_k)r_{k|k-1}. \end{aligned} \tag{2.74}$$

We can now see that the equation corresponds to the preposition in Theorem 2.3. □

Chapter 3

Multiple target tracking

So far, we have concerned ourselves with a single target, born at an arbitrary point in time, being detected by our sensor with a certain probability at each timestep and can be declared non-existent at a later time if, at enough consecutive timesteps a measurement will be missed. We are now also able to track a target inside the clutter of false measurements and handle situations of multiple measurements falling into the vicinity i.e., validation region of the given target.

Advancing further in this chapter, we will concern ourselves with a scenario with n_k targets and m_k measurements. When targets are well separated, a simple PDA filter or other single target tracking filter is usually sufficient. However, in the case of targets that appear close to one another, i.e., a measurement falls into an intersection of their validation regions, and a simple PDA filter approach begins to give sub-optimal results. Thus arises the necessity to calculate all possible combinations of associations between up to n_k targets and up to m_k measurements. Such a scenario may not be frequent, but it can lead to a significant computational complexity increase compared to simple PDA filter. Compared to other multiple target tracking filters, e.g., MHT, PHD, PMBM, the JPDA filter is still considered a reasonable compromise between computational complexity and robustness. [16]

3.1 PDA based filters

The most straightforward approach to tracking multiple targets would be to run multiple PDA filters (one for each target) independently. As mentioned earlier, this approach yields sub-optimal results if two targets cross paths or even start to follow a similar path. Since both PDA filters of each target work independently, they would gradually begin to ”merge” by incorporating measurements of both targets into their update step. Not only would the state estimate appear between the two targets, but by later possible splitting of the

targets, we would lose the track history of one of the two targets, since a new PDA filter would have to be initialized to track it again. The JPDA extension of the filter does offer greater robustness in this scenario by computing the β-coefficients jointly.

3.1.1 JPDA

Before we can start to derive the JPDA itself, several assumptions about the scenario and multi-target tracking model have to be made:

1. The number of targets in the surveillance region is constant and known.
2. Surviving targets motion is given by $f(x_k|x_{k-1})$.
3. A target with state x_k generates a measurement z_k with probability P_D.
4. The target motion and measurement model are Gaussian-linear.
5. The posterior densities of targets are independent and Gaussian

$$p^t_{k|k}(x^t_k) = \mathcal{N}(x^t_k; \hat{x}^t_{k-1}, P^t_{k-1}). \tag{3.1}$$

6. The clutter Poisson process has constant intensity λ.
7. The measurement of the detected target is related to its state by $f(z_k|x_k)$.

3.1.2 Prediction and update

Since the JPDA filter is built on the PDA filter, its prediction and update steps for every target state in the surveillance region are essentially the same as those for the PDA filter. The prediction step uses the Chapman-Kolmogorov Equation (2.39), and the update is given by Equation (2.55).

3.1.3 Joint probability data association coefficients

The main difference from the PDA filter is how β coefficients are defined and calculated.

First, let us introduce the association hypothesis as a vector

$$a_k = [a^1_k, a^2_k, \ldots, a^n_k], \tag{3.2}$$

where

$$a^t_k = \begin{cases} j & \text{if measurement j is assigned to target t,} \\ 0 & \text{if no measurement is assigned to target t,} \end{cases} \tag{3.3}$$

and if $a_k^s = a_k^l \neq 0$, then $s = l$, i.e. associations in a single hypothesis are mutually exclusive. Given this assumption, the total probability theorem yields

$$p(\mathbf{x}_k^1, \mathbf{x}_k^2, \ldots, \mathbf{x}_k^n | Z_{1:k}) = \sum_{a_k} p(\mathbf{x}_k^1, \mathbf{x}_k^1, \ldots, \mathbf{x}_k^n | a_k, Z_{1:k}) Pr\{a_k | Z_{1:k}\}, \tag{3.4}$$

The first term $p(\mathbf{x}_k^1, \mathbf{x}_k^2, \ldots, \mathbf{x}_k^n | a_k, Z_{1:k})$ is easily obtainable. If no measurement has been assigned to a target, then the predicted density $p_{k|k-1}^t(x_k^t)$ is used. Otherwise, if measurement $\mathbf{z}_k^{a_k^t}$ has been assigned to the target t, we update the predicted density using the Kalman Filter update step, giving us joint event-conditional posterior

$$p(\mathbf{x}_k^1, \mathbf{x}_k^2, \ldots, \mathbf{x}_k^n | a_k, Z_{1:k}) \propto \prod_{t:d_k'=0} p_{k|k-1}^t(\mathbf{x}_k^t) \prod_{t:_k'>0} f_{\mathbf{z}}(\mathbf{z}_k^{a_k'} | \mathbf{x}_k^t) p_{k|k-1}^t(\mathbf{x}_k^t). \tag{3.5}$$

Since we are assuming a linear Gaussian model, we can also express the posterior as

$$p(\mathbf{x}_k^1, \mathbf{x}_k^2, \ldots, \mathbf{x}_k^n | a_k, Z_{1:k}) = \prod_{t=1}^{n} \mathcal{N}(\mathbf{x}_k^t; \hat{\mathbf{x}}_k^{t,a_k^t}, \mathbf{P}_k^{t,a_k^t}). \tag{3.6}$$

In the next step, we need to obtain the probability $Pr\{a_k | Z_{1:k}\}$. Let us define track likelihood under hypothesis a_k as

$$l^{t,a_k^t} = \int f_{\mathbf{z}}(\mathbf{z}_k^{a_k^t} \mid \mathbf{x}_k^t) p_{k|k-1}^t(\mathbf{x}_k^t) \mathrm{d}\mathbf{x}_k^t = \mathcal{N}(\mathbf{z}_k^{a_k^t}; \hat{\mathbf{z}}_k^{a_k^t}, S_k^{a_k^t}). \tag{3.7}$$

By separating the current measurements from historical ones, we get

$$\begin{aligned} \Pr\{a_k | Z_{1:k}\} &= \Pr\{a_k | Z_{1:k-1}, Z_k, m_k\} && (3.8) \\ &\propto p(Z_k | m_k, a_k, Z_{1:k-1}) \Pr\{a_k | m_k, Z_{1:k-1}\}, && (3.9) \end{aligned}$$

where, according to [16], the first term can be expressed as

$$\begin{aligned} p(Z_k|m_k, a_k, Z_{1:k-1}) &= \int p(Z_k|m_k, a_k, \mathbf{x}_k^1, \ldots, \mathbf{x}_k^n) \\ &\quad \times p(\mathbf{x}_k^1, \ldots, \mathbf{x}_k^n|a_k, Z_{1:k-1}) \mathrm{d}\mathbf{x}_k^1 \cdots \mathrm{d}\mathbf{x}_k^n \\ &= \frac{1}{V\boldsymbol{\varphi}_k} \int \cdots \int \prod_{t:a_k^t>0} \left[f_{\mathbf{z}}(\mathbf{z}_k^{a_k^t} \mid \mathbf{x}_k^t) p_{k|k-1}(\mathbf{x}_k^t \mid Z_{1:k-1}) \right] \\ &\quad \times \prod_{t:a_k^t=0} \left[p_{k|k-1}(\mathbf{x}_k^t|Z_{1:k-1}) \right] \mathbf{d}\mathbf{x}_k^1 \cdots \mathbf{d}\mathbf{x}_k^n \\ &= \frac{1}{V\boldsymbol{\varphi}_k} \prod_{t:a_k^t>0} l^{t,\boldsymbol{a}_k^t}, \end{aligned} \tag{3.10}$$

where φ_k is number of clutter measurements under the k-th hypothesis.

To express the second term in 3.8 we will first need to track-wise detection event $\boldsymbol{\tau}(t)$ as a vector

$$\boldsymbol{\tau} = [\boldsymbol{\tau}(\mathbf{1}), ..., \boldsymbol{\tau}(n)], \tag{3.11}$$

$$\boldsymbol{\tau}(t) = \begin{cases} 1 & \text{if } a_k^t > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{3.12}$$

For any track, the probability of a single detection is given by constant parameter P_D; therefore

$$P\{\boldsymbol{\tau}(t)\} = \begin{cases} P_{\mathrm{D}} P_G & \text{if target is detected,} \\ 1 - P_{\mathrm{D}} P_G & \text{if target is not detected.} \end{cases} \tag{3.13}$$

Therefore the probability of event $\boldsymbol{\tau}$ can be written as product of individual detection event $\boldsymbol{\tau}(t) \in \boldsymbol{\tau}$ probabilities:

$$Pr\{\boldsymbol{\tau}\} = \prod_{t:a_k^t=0} (1 - P_D P_G) \prod_{t:ak^t>0} P_D P_G. \tag{3.14}$$

Further probabilities conditional on $\boldsymbol{\tau}$ are given as:

$$\Pr\{m_k|\tau\} = e^{-V\lambda} \frac{(V\lambda)^{\phi_k}}{\varphi_k!}, \tag{3.15}$$

$$\Pr\{a_k|\tau, m_k\} = \frac{\varphi_k!}{m_k!}. \tag{3.16}$$

The first equation holds because not included in $\boldsymbol{\tau}$ are Poisson distributed with the rate of $V\lambda$, and the total number of measurements m_k are uniquely given by $\boldsymbol{\tau}$ and $varphi_k!$. The second equation holds because for given $\boldsymbol{\tau}$, we have $m_k!/(m_k - \sum_t^1 \tau^t)! = m_k!/\boldsymbol{\varphi}_k!$ of equally probable permutations of

detections, with each permutation constituting distinct association hypothesis [16].

By rewriting the $Pr\{a_k|m_k\}$ into a more suitable form using all three equations above, we get

$$\begin{aligned}\Pr\{a_k|m_k\} &= \Pr\{a_k, \tau|m_k\} \\ &\propto \Pr\{a_k|\tau, m_k\}\Pr\{m_k|\tau\}\Pr\{\tau\} \\ &\propto (V\lambda)^{\varphi_k}\prod_{t:a_k^t=0}(1-P_{\mathrm{D}}^t P_G^t)\prod_{t:a_k^t>0}P_{\mathrm{D}}^t P_G^t. \end{aligned} \tag{3.17}$$

If we now substitute Equations 3.10 and 3.17 into 3.8 we get the final form of probability of association hypothesis

$$\begin{aligned}\Pr\{a_k|Z_{1:k}\} &\propto p(Z_k|m_k, a_k, Z_{1:k-1})\Pr\{a_k|m_k, Z_{1:k-1}\} \\ &\propto \frac{1}{V^{\varphi_k}}\prod_{t:a_k^t>0} l^{t,a_k^t}(V\lambda)^{\varphi_k}\prod_{t:a_k^t=0}(1-P_{\mathrm{D}}^t P_G^t)\prod_{t:a_k^t>0}P_{\mathrm{D}}^t P_G^t \\ &\propto \lambda^{\varphi_k}\prod_{t:a_k^t>0} l^{t,a_k^t}P_{\mathrm{D}}^t P_G^t\prod_{t:a_k^t=0}(1-P_{\mathrm{D}}^t P_G^t). \end{aligned} \tag{3.18}$$

As in the PDA filter, our goal is to merge all hypotheses into a single one, so we leave only a single Gaussian posterior distribution for each track. Having derived the probability of a whole assignment vector a_k, rather than it, we are interested in the marginal association probability $\beta_k^{t,i} = Pr\{a_k^t = i|Z_{1:k}\}$ for each measurement. Thus, we arrive at

$$\beta_k^{t,i} = Pr\{a_k^t = i|Z_{1:k}\} = \sum_{a_k:a_k^t=i} Pr\{a_k|Z_{1:k}\}. \tag{3.19}$$

Once we have these joint associations probabilities, we update each track t in the manner of PDA filter.

$$p_k^t(\mathbf{x}_k^t) = \sum_{j=0}^{m_k}\beta_k^{t,j}\mathcal{N}(\mathbf{x}_k^t; \hat{\mathbf{x}}_k^{t,j}, \mathbf{P}_k^{t,j}) \approx \mathcal{N}(\mathbf{x}_k^t; \hat{\mathbf{x}}_k^t, \mathbf{P}_k^t). \tag{3.20}$$

3.1.4 JIPDA filter

Having derived IPDA and JPDA filters, combining these two approaches is relatively straightforward. As in transition from PDA to IPDA filter, we assume that each target exists with the probability r_{k-1}^t at timestep $k-1$ and it continues to exist at timestep k with the probability P_s or ceases to exists with probability $1-P_s$. If it does not exist at timestep $k-1$, it will also not exist at timestep k.

The workflow loop in JIPDA filter is identical to the JPDA filter. The only difference is that the target's existence is not guaranteed; thus, we must

incorporate such a possibility into the JPDA detection event 3.14. Having the existence probability r_k^t of target t, we incorporate the probability into the equation:

$$Pr\{\boldsymbol{\tau}\} = \prod_{t:a_k^t=0} (1 - P_D P_G r_{k|k-1}^t) \prod_{t:ak^t>0} P_D P_G r_{k|k-1}^t. \tag{3.21}$$

Reflecting this adjustment into equation 3.18, we get the final form of association hypothesis probability for JIPDA

$$\Pr\{a_k|m_k\} \propto (V\lambda)^{\varphi_k} \prod_{t:a_k^t=0} (1 - P_{\mathrm{D}}^t P_G^t r_{k|k-1}) \prod_{t:a_k^t>0} P_{\mathrm{D}}^t P_G^t r_{k|k-1}. \tag{3.22}$$

3.1.5 Strengths and weaknesses of PDA, JPDA, and JIPDA filters

Even though we have managed, through gradual modification, to solve the problem of tracking a single target, single target in clutter, tracking multiple targets in clutter, and detection and termination of tracking for appearing and disappearing targets the JPDA and JIPDA filter suffer from several weaknesses that may drastically impact their performance.

Since the central idea of PDA and JPDA filters is to merge all possible posteriors into a single Gaussian posterior, also called *single-scan* method, does not allow to correct possible incorrect assignment in a previous step and inevitably causes information loss, resulting in lower robustness and precision compared to other approaches to target tracking.

Another problem of the JPDA filter is called *track coalescence*. Let us imagine a scenario where two targets are situated close to each other and moving in the same direction. Since the PDA and JPDA filters are based on averaging, the estimated target positions will, in time, merge and appear between the two real targets as in Figure 3.1.

Last but not least, the number of JIPDA hypotheses is subject to exponential growth, and despite the introduction of gating and track clustering (by overlapping validation regions), computing all possible association hypotheses can become unfeasible without additional optimization. For example, one possibility is to use Murthy's algorithm to find M best association hypothesis to compute β-coefficients. This, however, hinders the filter's performance further as we introduce another approximation to the model.

Nevertheless, compared to other filters, such as MHT or RFS-based filters, the JIPDA approach offers a computationally cheap and relatively robust solution to the target tracking problem.

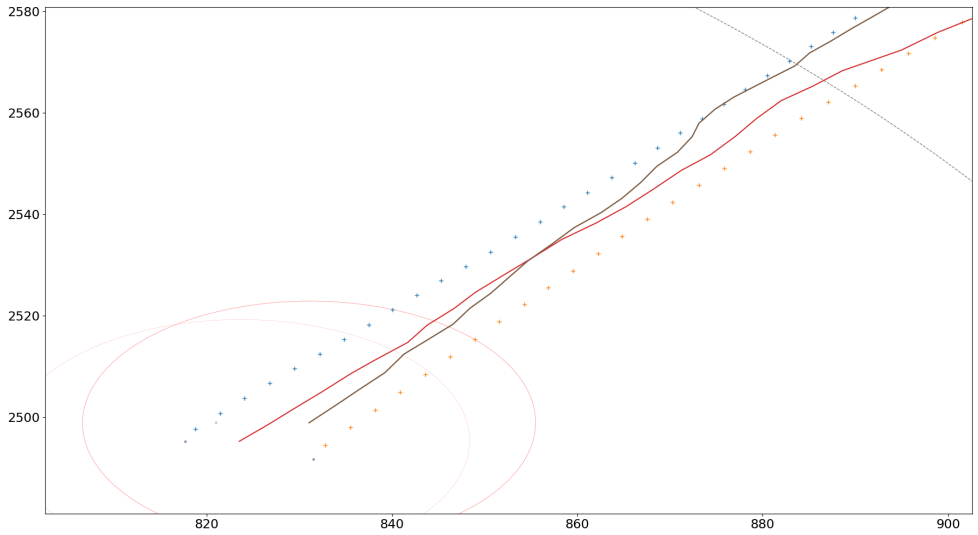


Figure 3.1 An example of track coalescence: Tracks slowly converge to each other and swap.

3.2 MHT filter and RFS based filters

As stated in the previous chapter, the JPDA filter belongs to the *single-scan* group of tracking filters. It reduces all possible posterior distributions into a single one for every target. This keeps the computational complexity relatively low at the expense of lower robustness and precision compared to the *multi-scan* group of filters.

One of the most basic representatives of the *multi-scan* group is the multiple hypothesis tracker (MHT) [15], [33], which creates multiple children assignment hypotheses from the parent – one for every measurement inside the validation region – and updates the underlying Kalman filter accordingly. This, however, leads to exponential growth of the number of hypotheses. Therefore, it is necessary to prune the least probable hypotheses [16].

The most modern approach to target tracking has been developed using random finite set (RFS) statistics, where the multi-object PDF is described as a function of RFS. The probability hypothesis density (PHD) filter [18], multi-Bernoulli mixture (MBM) filter [34], and Poission multi-Bernoulli mixture (PMBM) filter [35] are examples of such approaches. These filters offer greater robustness and accuracy in more complex scenarios [36]. However, this comes at the cost of far higher computational complexity than the JPDA approach. For this reason, they are out of the scope of this thesis and won't be discussed into more detail.

Chapter 4

Distributed modeling

Tracking objects using an array of different sensors is an important field of work in areas such as robotics and autonomous driving or the military [37], [38]. Combining data from multiple sensors can improve the model's accuracy, introduce redundancy in case of a sensor failure, or achieve the same performance using cheaper or smaller sensors.

The problem of cooperation between multiple agents or sensors can be approached using a centralized or fully distributed approach. In the former approach, every sensor node is processed by an agent, e.g., Kalman, PDA, JPDA filter, etc. Information from each node is then sent to a central node called *fusion center*, where it is merged to gain a final state estimate. The latter approach does not introduce a fusion node. Instead, each node communicates directly with every neighbor and incorporates their measurements into its estimate. This approach gives rise to multiple different approaches which can impose restrictions on the node network and put forth various ways to merge the additional information into each node's estimate.

In this chapter, we will introduce some of the approaches to distributed modeling and data fusion, focusing mainly on the diffusion approach.

4.1 Centralized approach

Centralized architecture is one of the most straightforward solutions to the data fusion problem. Data from each sensor is processed by an agent, e.g., Kalman filter, PHD filter, JIPDA filter The state estimate from each agent is then sent into a single fusion node. The fusion node is then responsible for track-to-track association and fusion of the incoming estimates. Even though the main advantages of the centralized approach are its simplicity and accuracy, it does introduce a single point of failure: the fusion center itself [22]. In addition, the fusion node requires more resources for processing information and high bandwidth for data transmission. Figure 4.1 shows examples of centralized architectures.

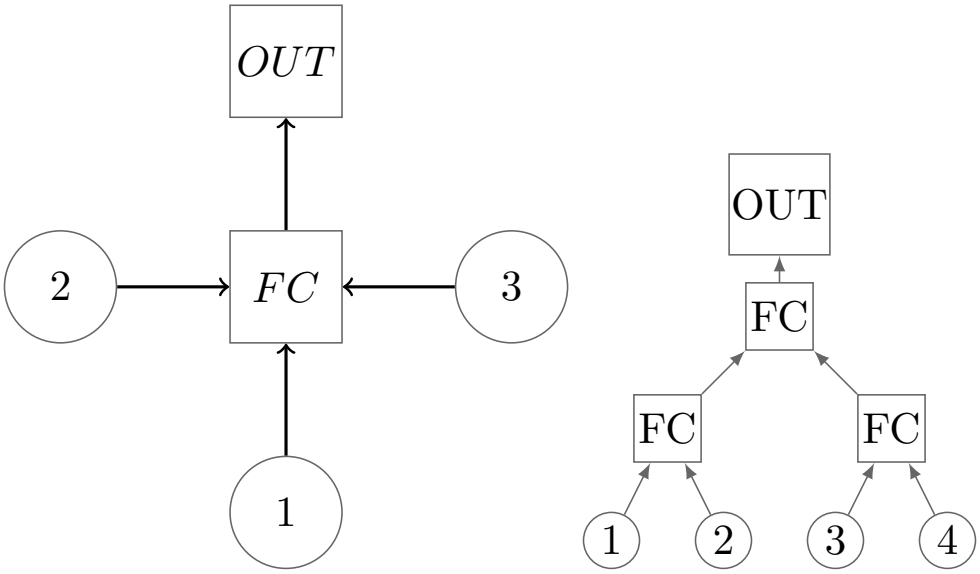


Figure 4.1 Fully centralized and hierarchical examples of centralized architectures with fusion centers

4.2 Distributed approach

The distributed approach aims to mitigate some of the problems of the centralized approach by delegating information fusion to agents themselves. While every agent the result of information fusion in every agent usually yields sub-optimal results – since not all agent nodes may be directly connected – it offers higher resilience to a failure.

4.2.1 Incremental strategy

The incremental strategy requires nodes to form a cyclic path (Hamiltonian cycle). Every timestep propagates the information to the next node on the

path until every node has been visited. While this strategy is relatively simple, it does suffer from several drawbacks. First, it introduces additional iterative steps to every scan cycle, increasing processing time at every timestep. Second, it limits the node-to-node connectivity to a single processing node on the path. Third, it is sensitive to link failures, requiring redesign of the network, which is generally an NP-hard problem [22].

4.2.2 Consensus strategy

The consensus strategy does not impose such strict limitations on the network structure. We expect the network to be a directed or undirected connected graph, with nodes of varying degrees – usually higher than one. In its basic form, the strategy consists of two steps: data acquisition for every node and collaborative data processing. Every node tries to iteratively find a convex combination of its and its neighbor's estimates and reach a consensus in the computation among them. One of the drawbacks of the consensus approach is its ability to perform real-time estimation. The iterative nature of the consensus computation mainly causes this. Even though implementations operating in a single step were proposed [26], [39] shows that these implementations may underperform compared to diffusion strategies and can become unstable in their estimation. Therefore, the focus will primarily be on the diffusion approach.

4.2.3 Diffusion strategy

In the diffusion approach, the exchange of information happens between neighboring nodes. Each node receives information from its neighbors and incorporates this additional data into its estimate. This way, the information slowly diffuses throughout the network. Since there is no need to reach a consensus with neighboring nodes, the computational complexity remains lower than that of consensus-based alternatives. The algorithm consists of two independent phases: *adaptation* and *combination*. This independece gives rise to four diffusion schemes *Adaptation*-only, *Combination*-only, *Adapt*-then-*Combine*, *Combine*-then-*Adapt* [22].

Before we describe the diffusion approach for the JIPDA filter itself. Let us now describe concepts of adaptation and combination in a general manner by returning to the concepts of conjugate priors, exponential family of distributions with a stochastic model determined by a fixed parameter θ.

For the following sections, let us represent the sensor network as an undirected connected graph $\mathcal{G} = (\mathcal{U}, \mathcal{V})$, where $\mathcal{U}$ represents set of nodes i.e. sensors and $\mathcal{V}$ represents connections between them. As every node communicates only with its direct neighbors, i.e., neighbors within jump distance of one, let us also denote a node as u and its neighborhood $\mathcal{U}_u = \{1, 2, 3, \ldots, u\}$. Note that by definition $u \in \mathcal{U}_u$.

4.2.3.1 Adaptation phase

The goal of the adaptation phase is to increase the amount of information the filter receives. This is achieved by acquiring relevant measurements from neighboring nodes and updating their prior estimates with them. In Bayesian terms, this equals to performing a batch update

$$p^u_{k|k}(\theta|\Delta^u_k) = p^u_{k|k-1}(\theta|\Delta^u_{k-1}, \bar{z}^u_k) \tag{4.1}$$

$$= p^u_{k|k-1}(\theta|\Delta^u_{k-1}) \prod_{u' \in \mathcal{U}} [f(z^u_k|\theta)]^{c^{u,u'}_k}, \tag{4.2}$$

where Δ^u_k is all the information available to node u at time k, including its past observations and all of its neighbor's past observations. The bar notation refers to all new observations available to node u at time k. $c^{u,u'}_k \in \{0, 1\}$ are the adaptation weights assigned by the node u to node u'. $c^{u,u'}_k = 1$ if the observation z^u_k is not an outlier; $c^{u,u'}_k = 0$ otherwise.

According to [26], under the conjugacy, the Bayesian update translates into the update of hyperparameters in the following way:

$$\xi^u_k = \xi^u_{k-1} + \sum_{u' \in \mathcal{U}} c^{u,u'}_k T(z^{u'}_k), \tag{4.3}$$

$$\nu^u_k = \nu^u_{k-1} + \sum_{u' \in \mathcal{U}} c^{u,u'}_k. \tag{4.4}$$

4.2.3.2 Combination phase

The goal of the combination phase is to share and improve the estimates of nodes. Every node receives current estimates from its neighbors, and *optimally* merges with its estimate. The posterior distributions used in this phase differ depending on whether we choose the ATC or CTA scheme. In case of the former the distributions are represented as $p^u_{k|k}(\theta|\Delta^u_k)$ received from Equation (4.1), or $p^u_{k|k-1}(\theta|\Delta^u_{k-1})$ in the latter case.

To merge these estimates, we need some "measure" of optimality. In the Bayesian theory, the most often used measure is the Kullback-Leilbler divergence [26], even though it does not satisfy the mathematical definition of a measure. We are trying to the final density $\tilde{p}^u_{k|k}(\theta|\Delta^u_k)$ (alternatively $\tilde{p}^u_{k|k-1}(\theta|\Delta^u_{k-1})$), whose divergence from densities $p^{u'}_{k|k}(\theta|\Delta^{u'}_k)$ ($p^{u'}_{k|k-1}(\theta|\Delta^{u'}_{k-1})$) is minimal. In mathematical terms, we are trying to minimize the cumulative loss

$$\sum_{u' \in \mathcal{U}_u} a^{u,u'} \mathcal{D}(\tilde{p}^u_{k|k}(\theta|\Delta^u_k)||p^{u'}_{k|k}(\theta|\Delta^{u'}_k). \tag{4.5}$$

$$\sum_{u' \in \mathcal{U}_u} a_k^{u,u'} D\Big(\tilde{p}_{k|k}^u(\theta) \,\Big\|\, p_{k|k}^{u'}(\theta)\Big) = \sum_{u' \in \mathcal{U}_u} a_k^{u,u'} \mathbb{E}_{\tilde{p}_{k|k}^u} \left[log \frac{\tilde{p}_{k|k}^u(\theta)}{p_{k|k}^{u'}(\theta)} \right] \tag{4.6}$$

$$= \mathbb{E}_{\tilde{p}_{k|k}^u} \left[log \frac{\tilde{p}_{k|k}^u(\theta)}{c \prod_{u' \in \mathcal{U}_u} \left[p_{k|k}^{u'}(\theta) \right]^{a_k^{u,u'}}} \right] \tag{4.7}$$

$$= D\left(\tilde{p}_{k|k}^u(\theta) \,\middle\|\, \prod_{u' \in \mathcal{U}_u} \left[p_{k|k}^{u'}(\theta) \right]^{a_k^{u,u'}} \right) \tag{4.8}$$

$$= \mathbb{E}_{\tilde{p}_{k|k}^u} \left[\log \frac{\tilde{p}_{k|k}^u(\theta)}{c \prod_{u' \in \mathcal{U}_u} \left[p_{k|k}^{u'}(\theta) \right]^{a_k^{u,u'}}} \right] \tag{4.9}$$

According to [26], using the definition of Kullback-Leilbler divergence, the convexity of $-\log(\cdot)$, and Jensen's inequality, we follow as:

$$= \mathbb{E}_{\tilde{p}_{k|k}^u} \left[-\log \frac{c \prod_{u' \in \mathcal{U}_u} \left[p_{k|k}^{u'}(\theta) \right]^{a_k^{u,u'}}}{\tilde{p}_{k|k}^u(\theta)} \right] \tag{4.10}$$

$$\geq -\log \mathbb{E}_{\tilde{p}_{k|k}^u} \left[\frac{c \prod_{u' \in \mathcal{U}_u} \left[p_{k|k}^{u'}(\theta) \right]^{a_k^{u,u'}}}{\tilde{p}_{k|k}^u(\theta)} \right] \tag{4.11}$$

$$= -log1 = 0 \tag{4.12}$$

We can see that the Kullback-Leilbler divergence is non-negative and zero if

$$\tilde{p}_{k|k}^u(\theta) = \prod_{u' \in \mathcal{U}_u} \left[p_{k|k}^{u'}(\theta) \right]^{a_k^{u,u'}} . \tag{4.13}$$

If we look at the solution at Equation (4.13), we can see that it is only a simple combination of conjugate priors. This allows us to translate this solution into the update of the hyperparameters as

$$\tilde{\xi}_k^u = \sum_{u' \in \mathcal{U}_u} a_k^{u,u'} \xi_k^{u'}, \tag{4.14}$$

$$\tilde{\nu}_k^u = \sum_{u' \in \mathcal{U}_u} a_k^{u,u'} \nu_k^{u'}, \tag{4.15}$$

for the ATC scheme and alternatively for the CTA scheme:

$$\tilde{\xi}_{k-1}^{u} = \sum_{u' \in \mathcal{U}_u} a_k^{u,u'} \xi_{k-1}^{u'}, \tag{4.16}$$

$$\tilde{\nu}_{k-1}^{u} = \sum_{u' \in \mathcal{U}_u} a_k^{u,u'} \nu_{k-1}^{u'}, \tag{4.17}$$

The ATC scheme can be finally described by the algorithm below.

Algorithm 2 ATC diffusion estimation

Require: Initilize nodes $u = 1, \ldots, |U|$ with prior densities $p_0^u(\theta)$

1: **for** $k \leftarrow 1, 2, 3, \ldots$ **do**

Adaptation phase:

2: Get observations $Z_k^{u'}$ from neighbors $u' \in \mathcal{U}_u$.

3: Acquire adaptation weights $c_k^{u,u'}$.

4: Update prior distribution of θ as in 4.1.

Combination Phase:

5: Acquire combination weights $a_k^{u,u'}$.

6: Get posterior densities from neighboring nodes $u' \in \mathcal{U}_u$.

7: Update hyperparameters of prior distribution of θ as in 4.14.

8: **end for**

▶ Note 4.1. We have not described a way to acquire adaptation and combination weights. This topic is outside of the scope of this thesis. Therefore, we further expect the weights to be uniform.

Chapter 5

Diffusion JIPDA filter

In the first chapter, we introduce and derive the Kalman filter. Later, we expanded on the foundation of the Kalman filter and extended it to work in a cluttered environment (PDA filter) and track multiple targets (JPDA). Collaterally, we have also described and derived a way to handle appearing and disappearing targets. Lastly, we have laid the foundation for collaborative filtering in a more general manner. In this chapter, we will combine the results from the previous chapter and propose a novel collaborative method based on the diffusion framework and JIDPA filter.

5.1 Review of the JIPDA filter

First, let us review the concepts we used in the previous chapters. We have a connected undirected graph $\mathcal{G} = (\mathcal{U}, \mathcal{V})$, where $\mathcal{U} = 1, 2, \ldots, |\mathcal{U}|$. Each node u observes an environment with n_k^u targets present. At discrete timesteps $k = 0, 1, 2, \ldots$, the node receives a set of $m_k^u \geq 0$ measurements denoted $Z_k^u = \{\mathbf{z}_k^{u,1}, \mathbf{z}_k^{u,2}, \ldots, \mathbf{z}_k^{u,m_k^u}\}$, who may or may not originate from a target. Each target $t \in \mathbb{N}$ does generate at most one measurement per timestep. To track these targets, each node is also running an independent JIPDA filter, which estimates each target's hidden Markov model $\mathbf{x}_k^{u,t}$. We also expect that every target behaves independently according to an observation model $f(\mathbf{z}_k^{u,t}|\mathbf{x}_k^{u,t})$ with a probability density $p(\mathbf{x}_k^{u,t}|\mathbf{x}_{k-1}^{u,t})$. We model the clutter in the environment with volume V as a Poisson point process with intensity $\lambda > 0$. Targets themselves may not be detected at various times with probability $1 - P_D$ and may appear and disappear during the whole process at any time. For this reason, we also track their probability of existence $r_k^{u,t} \in [0, 1]$, which behaves according to Markov-type model $P(r_k^{u,t}|r_{k-1}^{u,t}) = r_{k-1}^{u,t} P_S$. $P_S \in [0, 1]$ is the survival probability of a target. It is modeled as invariant in time and is the same for all targets. Additionally, we do not expect to detect the target at every step. Therefore, we denote the probability of detection as $P_D \in (0, 1]$. If

the target is detected, we also expect the measurement itself to appear within a reasonable vicinity of the target. For this reason we also define $P_G \in (0, 1]$ as the gating probability, which was described in Section 2.5.4.

The Bayesian approach takes advantage of past knowledge, i.e., past observations $Z^u_{0:k-1}$. The prior distribution representing this knowledge can be expressed as

$$\begin{aligned} &p^u_{k-1|k-1}(\boldsymbol{x}^{u,1}_{k-1}, \ldots, \boldsymbol{x}^{u,n}_{k-1} | Z^u_{0:k-1}) \\ &\qquad = \prod_{t=1}^{n^u_k} p^u_{k-1|k-1}(\boldsymbol{x}^{u,t}_{k-1} | Z^u_{0:k-1}). \end{aligned} \tag{5.1}$$

The feedback loop at a time k consists of three steps: prediction, update, and merging.

First, we predict the state of the process model. This prediction follows the Chapman-Kolmogorov equation

$$p^u_{k|k-1}(\boldsymbol{x}^{u,1}_k, \ldots, \boldsymbol{x}^{u,n^u_k}_k | Z^u_{0:k-1}) \tag{5.2}$$

$$= \prod_{t=1}^{n^u_k} \int p(\boldsymbol{x}^u_k | \boldsymbol{x}^u_{k-1})\, p^u_{k-1|k-1}(\boldsymbol{x}^{u,t}_{k-1} | Z^u_{0:k-1}) \mathrm{d}\boldsymbol{x}^{u,t}_{k-1}. \tag{5.3}$$

Next, we update the existence probability of each target

$$r^{u,t}_{k|k-1} = P_S \cdot r^{u,t}_{k-1|k-1}. \tag{5.4}$$

Once we receive measurements, we perform the Bayesian update. Symbolically, that means

$$\begin{aligned} p^u_{k|k}(\boldsymbol{x}^{u,1}_k, \ldots, \boldsymbol{x}^{u,n^u_k}_k | Z^u_{0:k}) \propto\ & p^u_{k|k-1}(\boldsymbol{x}^{u,1}_k, \ldots, \boldsymbol{x}^{u,n^u_k}_k | Z^u_{0:k-1}) \\ & \times f(Z^u_k | \boldsymbol{x}^{u,1}_k, \ldots, \boldsymbol{x}^{u,n^u_k}_k). \end{aligned} \tag{5.5}$$

This, however, is only the conceptual form of the update. We must address the uncertainty of the associations between measurements and target states. To do so, we introduce the single association hypotheses

$$a^{u,t,h}_k = \begin{cases} j & \text{if target } t \text{ is associated with } \boldsymbol{z}^{u,j}_k, \\ 0 & \text{if target } t \text{ is misdetected.} \end{cases}$$

Note that, in these hypotheses, we already only consider measurements that do appear in in a reasonable distance from the predicted position, given by the parameter P_G. By grouping all association hypotheses for n_k targets, we can form association hypotheses vector

$$\boldsymbol{a}^{u,h}_k = [a^{u,1,h}_k, \ldots, a^{u,n^u_k,h}_k], \qquad h = 1, \ldots, H. \tag{5.6}$$

For each of these vector we assign a probability $P(\boldsymbol{a}_k^{u,h}|Z_{0:k-1}^u)$. It is then possible to construct a joint posterior distribution $p_{k|k}^u(\boldsymbol{x}_k^{u,1},\ldots,\boldsymbol{x}_k^{u,n_k^u}|\boldsymbol{a}_k^{u,h},Z_{0:k}^u)$ for each $a_k^{u,h}$. By marginalizing over all of the association hypotheses, we arrive at

$$p_{k|k}^u(\boldsymbol{x}_k^1,\ldots,\boldsymbol{x}_k^{n_k}|Z_{0:k}^u) = \sum_h p_{k|k}^u(\boldsymbol{x}_k^1,\ldots,\boldsymbol{x}_k^{n_k}|\boldsymbol{a}_k^h,Z_{0:k}) \times P(\boldsymbol{a}_k^h|Z_{0:k}^u), \tag{5.7}$$

where $P(\boldsymbol{a}_k^h|Z_{0:k}^u)$ are posterior probabilities of hypothesis weights

$$P(\boldsymbol{a}_k^{u,h}|Z_{0:k}^u) \propto \prod_{t:a_k^{u,t,h}=0}(1-P_DP_G) \times \prod_{t:a_k^{u,t,h}\neq 0}\frac{P_DP_G\int f(\boldsymbol{z}_k^j|\boldsymbol{x}_k^{u,t})p_{k|k-1}^u(\boldsymbol{x}_k^{u,t}|Z_{0:k-1}^u)\mathrm{d}\boldsymbol{x}_k^{u,t}}{\lambda}. \tag{5.8}$$

Note that posterior hypothesis probability is a product of misdetection probabilities, i.e., $a_k^{u,t,h}=0$ and associations $a_k^{u,t,h}\neq 0$ scaled by their likelihood. The first factor in (5.7) is a sum of the posterior distribution of target states given the hypothesis h

$$\begin{aligned} p_{k|k-1}^u(\boldsymbol{x}_k^1,\ldots,\boldsymbol{x}_k^{u,n_k}|\boldsymbol{a}_k^h,Z_{0:k}^u) &\propto \prod_{t:\boldsymbol{a}_k^{u,t,h}=0} p_{k|k-1}^u(\boldsymbol{x}_k^{u,t}|Z_{0:k-1}^u) \\ &\times \prod_{t:\boldsymbol{a}_k^{u,t,h}\neq 0} f(\boldsymbol{z}_k^j|\boldsymbol{x}_k^{u,t})p_{k|k-1}^u(\boldsymbol{x}_k^{u,t}|Z_{0:k-1}^u). \end{aligned} \tag{5.9}$$

Now that we have a form of the posterior distribution, we have to address one last common problem of multi-target tracking filters. That is, the number of factors in (5.7) can grow exponentially with time. To solve this issue, we marginalize over all of the possible association hypotheses of each target, resulting in

$$p_{k|k}^u(\boldsymbol{x}_k^{u,t}|Z_{0:t}^u) = \sum_{j=0}^{m_k^u}\beta_k^{t,j}p_{k|k}^u(\boldsymbol{x}_k^{u,t}|Z_{0:k}^u,\boldsymbol{z}_k^{u,j}), \tag{5.10}$$

where

$$\beta_k^{u,t,j} = \sum_{h:a_k^{u,t,h}=j} P(a_k^{u,t,h}|Z_{0:k}^u). \tag{5.11}$$

At last, we update the existence probabilities as

$$r_{k|k}^{u,t} = \frac{r_{k|k-1}^{u,t}\mathcal{L}_k^{u,t}}{(1 - r_{k|k-1}^{u,t}) + r_{k|k-1}^{u,t}\mathcal{L}_k^{u,t}}, \tag{5.12}$$

where

$$\mathcal{L}_k^{u,t} = (1 - P_D) + \frac{P_D \sum_{a_k^{u,t}=1}^{m_k^u} \int f(\boldsymbol{z}_k^j|\boldsymbol{x}_k^{u,t}) p_{k|k-1}^u(\boldsymbol{x}_k^{u,t}|Z_{0:k-1}^u)\mathrm{d}\boldsymbol{x}_k^{u,t}}{\lambda}. \tag{5.13}$$

Having now multiple densities for a single target – each for every measurement in the validation region – We need to merge them

$$\hat{p}_{k|k}(\boldsymbol{x}_k^{u,t}|Z_{0:t}^u) = \arg\min_{q(\boldsymbol{x}_k^{u,t})} D[p_{k|k}^u(\boldsymbol{x}_k^{u,t}|Z_{0:t})||q(\boldsymbol{x}_k^{u,t})], \tag{5.14}$$

5.2 Collaborative filtering

Let us now describe the collaboration part itself.

5.2.1 Adaptation phase

In this phase, every node u receives a set of measurements from every neighboring node u'. It could seem handy to combine them into a single set. Recalling model assumptions introduced in Section 2.5.1, we can see that this approach would violate the assumption 1. We must ensure that the Bayesian update contains, at most, one measurement of every target. Fortunately, the Bayesian theorem does allow us to update our knowledge about model parameters repetitively:

$$p_{k|k}^u(\boldsymbol{x}_k^{u,1}, \ldots, \boldsymbol{x}_k^{u,n_k^u}|Z_{0:k}^u) \propto p_{k|k-1}^u(\boldsymbol{x}_{u,k}^1, \ldots, \boldsymbol{x}_k^{u,n}|Z_{0:k-1}^u) \times \prod_{u' \in \mathcal{U}} f(Z_k^{u'}|\boldsymbol{x}_k^{u,1}, \ldots, \boldsymbol{x}_k^{u,n_k^u}). \tag{5.15}$$

In practice, the (5.15) turns into a series of sequential updates, one for every neighbor, where the resulting distribution serves as a prior distribution for the following update.

$$\underbrace{p_{k|k-1}^u(\circ|Z_{0:k-1}^u)}_{\text{predicted}} \xrightarrow[Z_k^{u_1}]{\text{update}} \cdots \xrightarrow[Z_k^{u_{|\mathcal{U}_u|}}]{\text{update}} \underbrace{p_{k|k}^u(\circ|Z_{0:k}^u)}_{\text{posterior}}.$$

5.2.2 Combination phase

In the combination phase, every neighbor u' sends their approximate posterior densities $\hat{p}^{u'}_{k|k}(\boldsymbol{x}_k^{u',1}, \ldots, \boldsymbol{x}_k^{u',n_k^{u'}} | Z^{u'}_{0:k})$ to node u, which then forms a mixture

$$\begin{aligned} &\tilde{p}^{u}_{k|k}(\boldsymbol{x}_k^{u,1}, \ldots, \boldsymbol{x}_k^{u,n_k^{u}} | \{Z^{u'}_{0:k}; u' \in \mathcal{U}_u\}) \\ &\quad = \frac{1}{|\mathcal{U}_u|} \sum_{u' \in \mathcal{U}_u} p^{u'}_{k|k}(\boldsymbol{x}_k^{u',1}, \ldots, \boldsymbol{x}_k^{u',n_k^{u'}} | Z^{u'}_{0:k}), \end{aligned} \tag{5.16}$$

where all components have uniform weights.

The newly formed mixture may now contain multiple marginal posterior densities belonging to a single target. Therefore, to not only reduce the number of components in the mixture but mainly to increase the accuracy of our predictions; we must merge similar components in the mixture. To do so, first, we have to determine which components are sufficiently similar to each other to be considered originating from a single target. For this purpose, we use a convenient measure of similarity, the Bhattacharyya distance

$$B\left[p, q\right] = -\log \sqrt{p(x) \cdot q(x)}, \tag{5.17}$$

where $p(x)$ and $q(x)$ are marginal distributions of the mixture.

The Bhattacharyya distance fully reflects the properties of compared distributions, not only their moments. Additionally, it is symmetric, which further supports its use in this application. We declare that two marginal distributions are identical if their Bhattacharyya distance is less than an arbitrarily defined tolerance $\alpha > 0$.

Finally, we merge all marginal densities declared identical in a Kullback-Leilbler sense. The existence probabilities are combined by averaging existence probabilities of relevant marginal densities. Two optimal – in Kullback-Leilbler sense – methods to fuse marginal densities that will be investigated in the next chapter are the moment matching method, which is also used in the PDA filter, and the covariance intersection as proposed in [26].

Algorithm 3 Diffusion JIPDA Algorithm

Require: The nodes $u = 1, \ldots, |\mathcal{U}|$ are initialized with the prior densities $p^u_{0|0}(\boldsymbol{x}^{u,t}|Z^t_0)$. The tolerance $\alpha > 0$ is set. For $k = 1, 2, \ldots$ and each node i do:

Prediction step:

Perform prediction – Equations (5.3), (5.4).

Adaptation phase:

for $u' \in \mathcal{U}$ **do**
  Get observations Z^u_k and $z^{u'}_k$ of neighbors $u \in \mathcal{U}_u$.
  Construct H association vectors, Eq. (5.6).
  Compute hypotheses probabilities, Eq. (5.8).
  Update states, Eq. (5.9).
  Calculate marginal probabilities, Eq. (5.11).
  Calculate marginal state densities, Eq. (5.10).
  Update existence probabilities, Eq. (5.12).
  For all targets, merge marginals, Eq. (5.14).
end for

Combination phase:

Get the posterior densities of neighbors $u \in \mathcal{U}_u$.
Detect common targets using Bhattacharrya distance (5.17).
Fuse common targets' posteriors.
Fuse existence probabilities.

Chapter 6

Numerical experiments

In this Chapter, we present the results of multiple simulation scenarios, which compare multiple schemes of node cooperation against a variant where no communication and cooperation between nodes takes place. Cooperation schemes, including the combine phase, also include the name of a method used to fuse marginal densities of the mixture described by Equation (5.16).

The cooperation schemes, fusion method used, and their respective abbreviations used in graphs in this chapter are following:

1. Adaptation-only (A)
2. Combination-only, moment matching (C_MM)
3. Combination-only, covariance intersection (C_CI)
4. Adapt-then-Combine, moment matching (ATC_MM)
5. Combine-then-Adapt, covariance intersection (CTA_CI)
6. No cooperation (NO_COOP)

All presented experiments are based on the CVM model in a 2D plane described in Example 2.1. We also assume that measurements arrive in constant intervals of 1 second, i.e., $\Delta k = 1$. The transition matrix F, process noise covariance matrix Q, measurement matrix H, and measurement noise covariance matrix R have the following form

$$F = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad Q = 0.001 \cdot \boldsymbol{I}_{4\times 4},$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \qquad R = \begin{bmatrix} 15 & 0 \\ 0 & 15 \end{bmatrix}.$$

The clutter follows the Poisson point process with intensity $\lambda = 10^{-5}$. The validation gate is set to 98% highest probability density region, and the probability of a target being detected $P_D = 0.8$. The JIPDA algorithm is initialized with the initial existence probablity $r_0^{u,t} = 0.9$, probability of survival $P_S = 0.95$ and survival threshold $P_T = (r_0^{u,t})^4$. Finally, the Bhattacharyya distance threshold is set to $\beta = 10$.

The performance of the proposed algorithm is measured using the generalized optimal sub-pattern assignment (GOSPA) metric [40].

▶ **Definiton 6.1.** *Let $c > 0, 0 < \alpha$ and $1 \leq p < \infty$. Let $d(x, y)$ denote a metric for any $x, y \in \mathbb{R}^N$ and let $d^{(c)}(x, y) = min(d(x, y), c)$, be its cut-off metric. Let Π_n be the set of all permutations of $\{1, \ldots, n\}$ for any $n \in \mathbb{N}$ and any element $pi \in \Pi_n$ be a sequence $(\pi(1), \ldots, \pi(n))$. Let $X = \{x_1, \ldots, x_{|X|}\}$ and $Y = \{y_1, \ldots, y_{|Y|}\}$ be a finite subsets of $\mathbb{R}^N$. For $|X| \leq |Y|$, the GOSPA metric is defined as*

$$
\begin{aligned}
& d_p^{(c,\alpha)}(X, Y) \\
& \triangleq \left(\min_{\pi \in \Pi_{|Y|}} \sum_{i=1}^{|X|} d^{(c)}(x_i, y_{\pi(i)})^p + \frac{c^p}{\alpha}(|Y| - |X|) \right)^{\frac{1}{p}}. \qquad (6.1) \\
& If |X| > |Y|, d_p^{(c,\alpha)}(X, Y) \triangleq d_p^{(c,\alpha)}(Y, X)
\end{aligned}
$$

However, to use GOSPA, we still have to choose the four parameters of the metric, namely, the distance metric $d(x, y)$ and parameters c, α, and p. Since the targets are moving through Euclidean space, a natural choice can be the Euclidean distance. Although, a different metric, e.g., root mean square error (RMSE), is also a viable option [40]. The parameter c, which determines the maximum allowable localization error, was set as $c = 10$. The parameter α, which is used to tune penalization due to cardinality mismatch, i.e., penalization for mismatch between real targets and detected targets, has been set to $\alpha = 2$ as [40] argues that this is the most appropriate choice for MTT algorithms. Lastly, the parameter p, used to tune outliers' penalization, has been set to $p = 2$. Higher value penalizes outliers more. In the following scenarios with multiple JPDA instances and radar, GOSPA is calculated for every JPDA instance separately with its own estimates. The final value of GOSPA for the whole network of agents is calculated as an average of these individual GOSPA values.

▶ Note 6.2. Since we are interested in the accuracy of the proposed algorithm, we will be using mainly the localization error part of the GOSPA metric. The metric's missed target and false detection parts have no meaning when we use IPDA.

6.1 Single target, two radars with unlimited FOV

We start with the simplest example. This scenario contains only a single target and two radars with unlimited FOV, shown in Figure 6.1. The lifespan of the target is 140 seconds. This test aims to test the effectiveness of the collaboration for a more extended period of time. After the initial period of a few timesteps, we should see the lower GOSPA value of cooperation variants compared to the no cooperation variant.

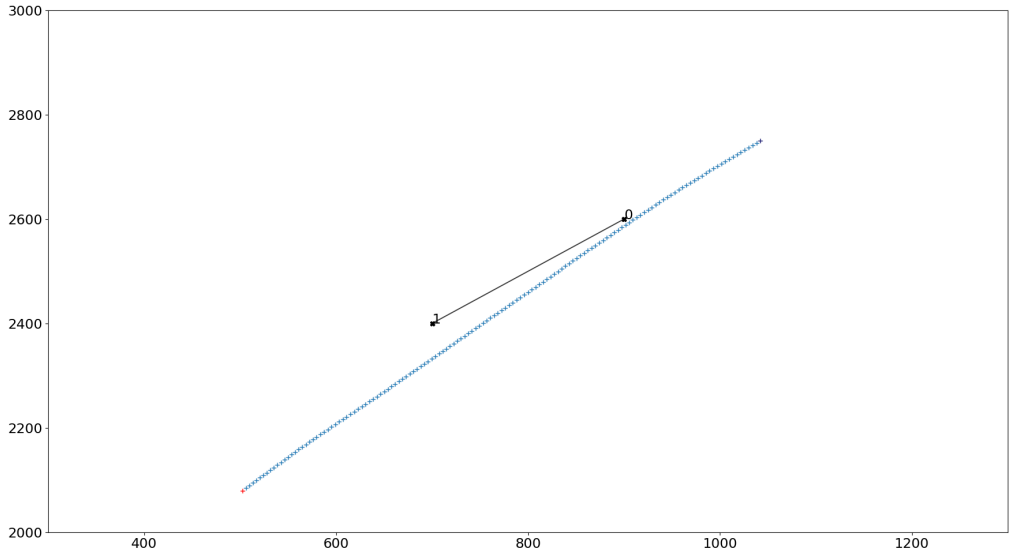


Figure 6.1 Trajector of a target in a simple scenario, two radars, the black line between them represents a communication link, clutter has been omited for clarity

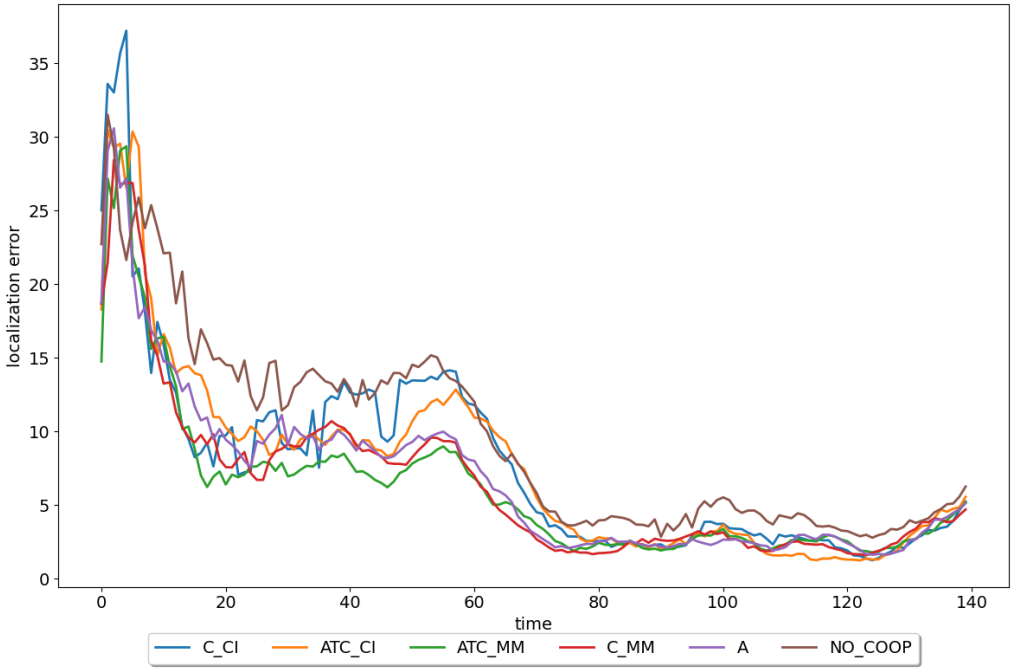


Figure 6.2 Averaged localization error of GOSPA in simple scenario

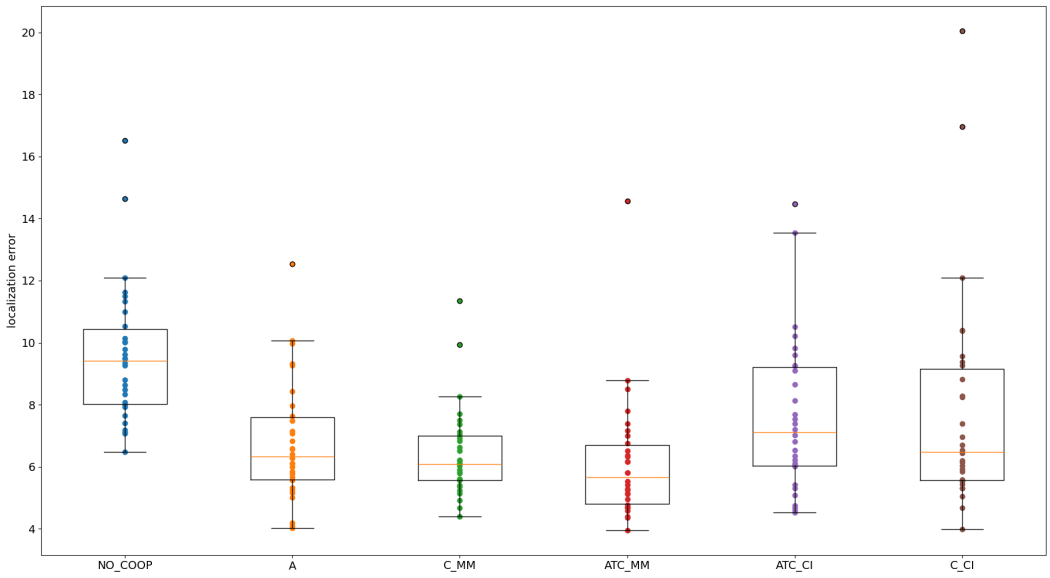


Figure 6.3 Comparison of averaged localization error of 30 simulation runs, one dot represents one simulation run.

As shown in Figure 6.2, the NO_COOP variant is quickly surpassed by all other variants after a few initial timesteps. At the $k = 57$ mark, the GOSPA

of all cooperation variants increases, most notably for covariance intersection variants. This could be caused by multiple clutter measurements appearing in the validation region. From Figure 6.3, we can see that the GOSPA averaged over the whole simulation are mostly lower than values of NO_COOP, even though the C_CI variant outliers have the highest values of all variants. For the Adaptation-only and both schemes with the moment matching method used, we can see the improved performance of the tracking algorithm.

6.2 Single target, two radars with limited FOV

The scenario in this example contains the same target and radars as the previous one. Now, however, the FOV of those radars was reduced. At the start and the end, the target is located outside the FOV of both radars; we should, therefore, see the localization error equal to zero. Once the target enters the FOV of the first radar (Figure 6.5), we should see similar results for both cooperation and no cooperation variants, as only a single radar sees the target. Then, we should see an improvement in the accuracy of cooperation variants once the target enters the intersection of FOVs of both radars (Figures 6.6, 6.7). This experiment aims to test information sharing between neighboring radars as a tracked target enters FOV the neighbor.

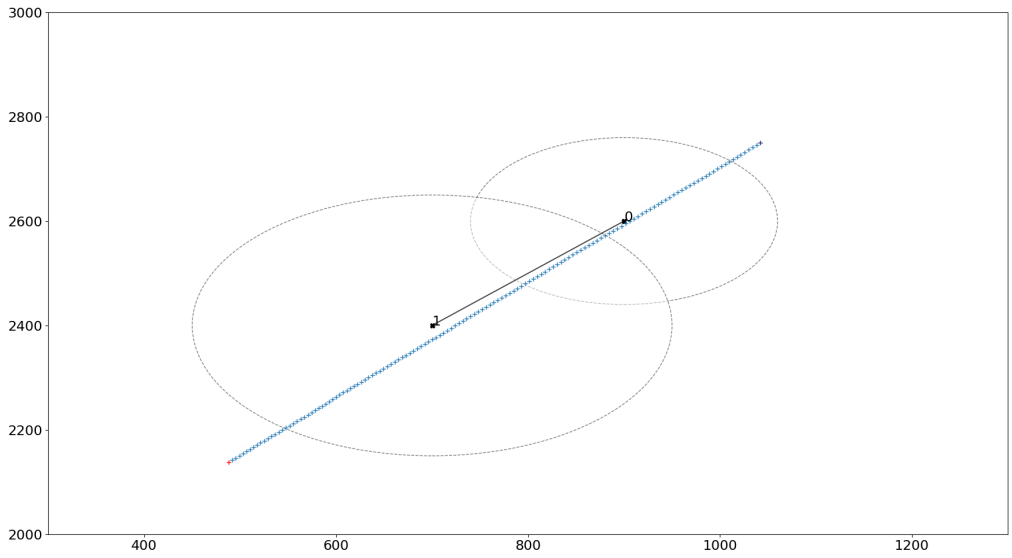


Figure 6.4 Trajectory of a target in a simple scenario with limited radar FOV

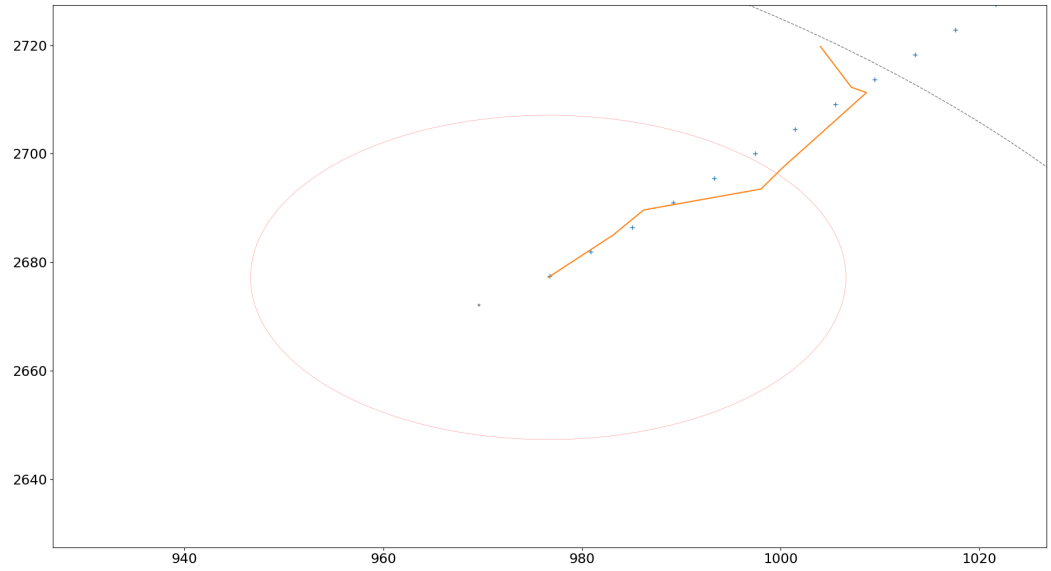


Figure 6.5 Target entering FOV of the first radar (border of the FOV is visible in right upper corner)

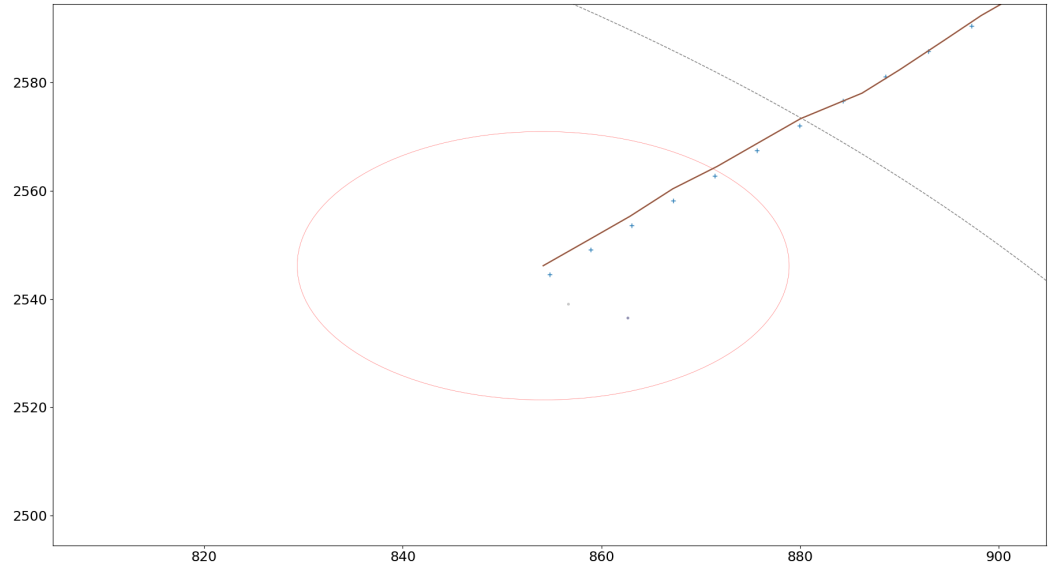


Figure 6.6 Target entered FOV of the second radar, simulation with ATC_MM scheme

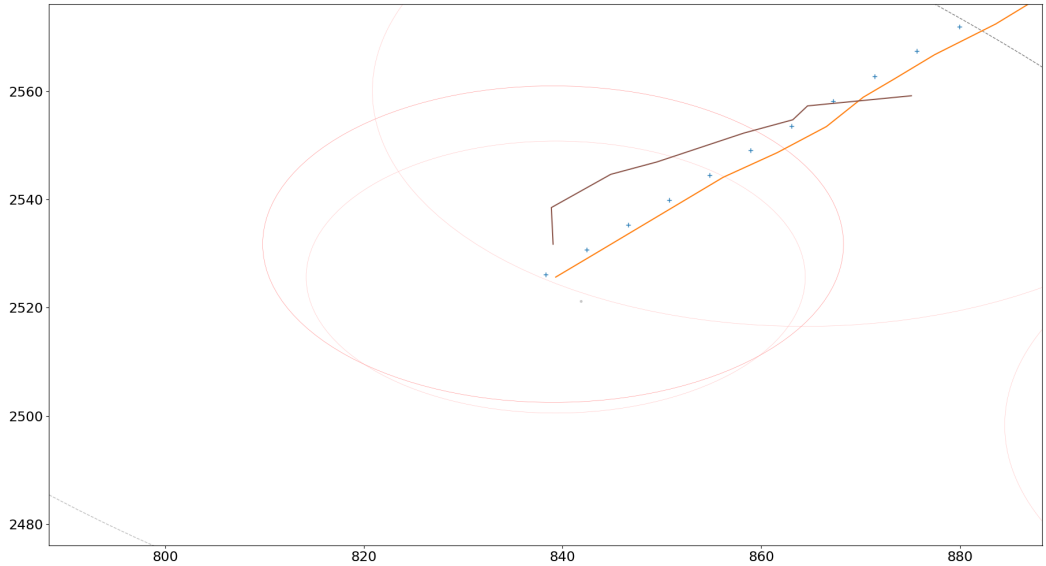
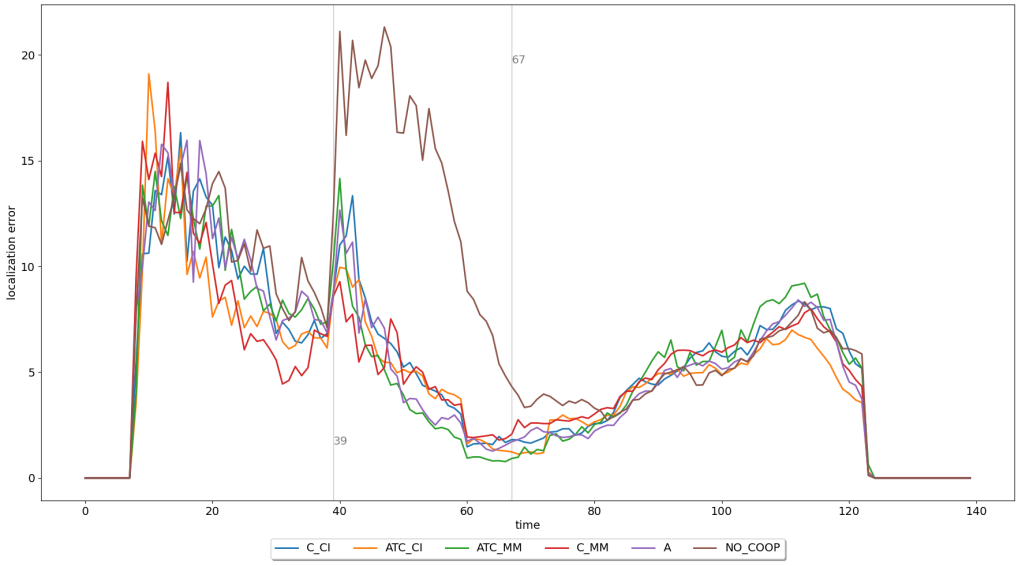


Figure 6.7 Target entering FOV of the second radar, simulation with NO_COOP scheme



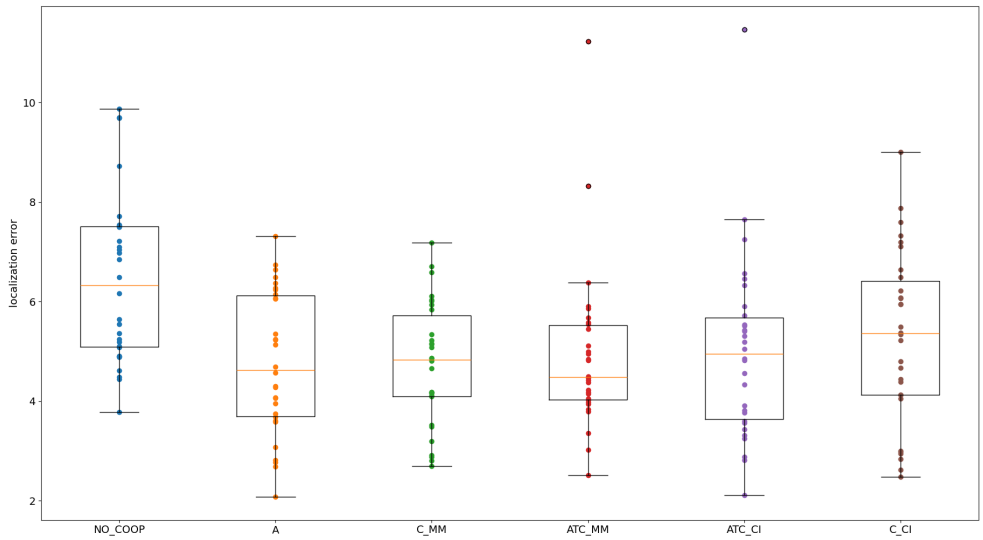


Figure 6.8 Comparison of averaged localization error of 30 simulation runs, one dot represents one simulation run.

As we can see in Figure 6.4, the value of GOSPA spikes sharply at $k = 39$ in all variants, most notably in the NO_COOP variant. All cooperation variants are able to quickly recover and track the target with significantly higher precision than the NO_COOP variant, while the target stays in the FOV intersection. Once the target leaves the intersection at $k = 67$, the value of GOSPA for all variants stays at similar values. This is expected as after the $k = 67$ mark; the target is again visible only to a single radar. Even though the target spent only a fifth of the time in the FOV intersection, we can see from Figure 6.8 that average values of all cooperation variants are still, on average, lower, although differences in accuracy are lower than in the previous example.

6.3 Two targets crossing path, two radars with limited FOV

In this scenario, we have the same radar and FOV layout as in the previous example. This time, two targets cross paths at the FOV intersection. Both targets fly in slightly similar directions to increase the time they spend close to each other. In this experiment, we aim to test the ability of the proposed filter to deal with the track coalescence problem and its performance in such a situation.

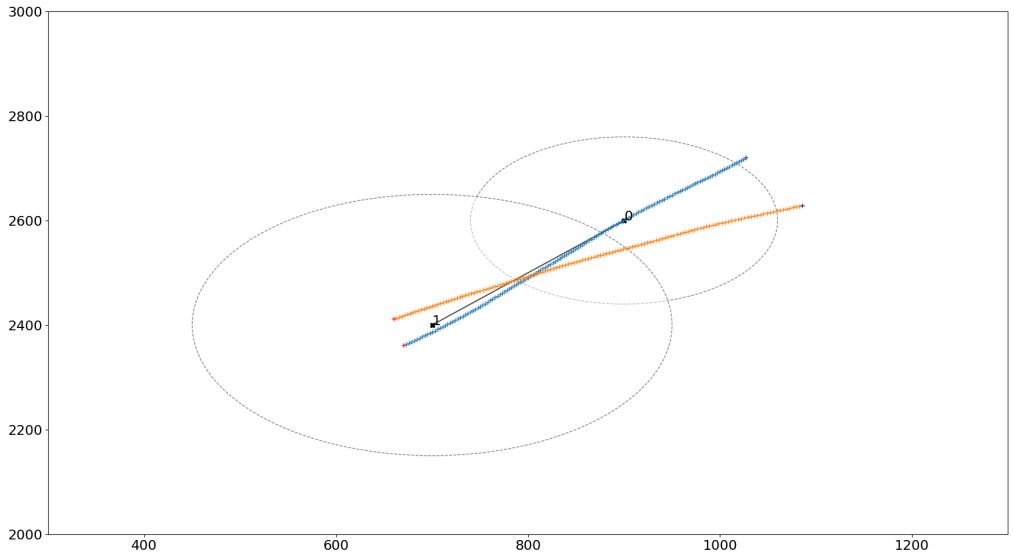


Figure 6.9 Scenario of two targets approaching each other in the FOV intesection

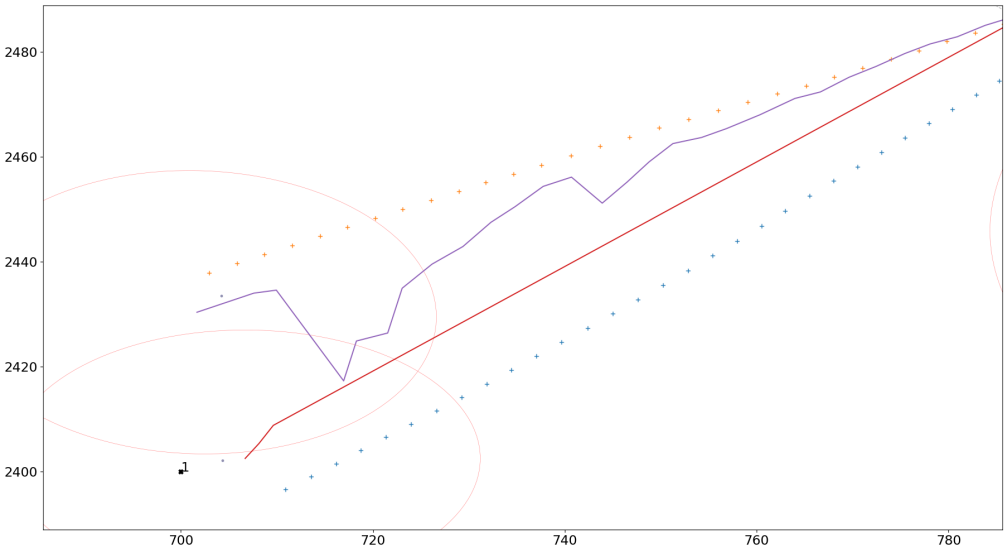


Figure 6.10 Zoomed view on the crossing region

Figure 6.11 Averaged GOSPA for the target crossing scenario

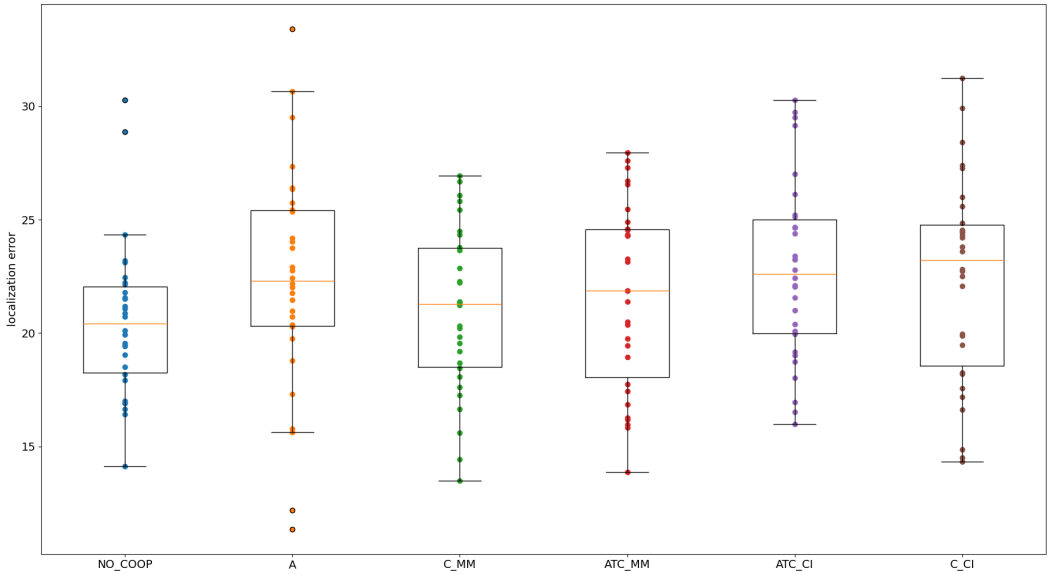


Figure 6.12 Comparison of averaged localization error of 30 simulation runs, one dot represents one simulation run.

In this scenario, we can see in both Figures 6.11 and 6.12 that the proposed algorithm fails in such a scenario. The NO_COOP variant performs, on average, better than all other methods. This can be caused by several factors. This can be caused by the validation gate to be set too low, increasing the possibility of measurement of the second target to be included in the adaptation

phase update of the first target and vice versa.

6.4 complex scenario

In this scenario, we have a network of 8 radars and multiple targets leaving and entering their FOVs. Targets do not cross cross paths and travel in straight lines. This experiment aims to test the overall performance of the proposed filter with a more extensive network.

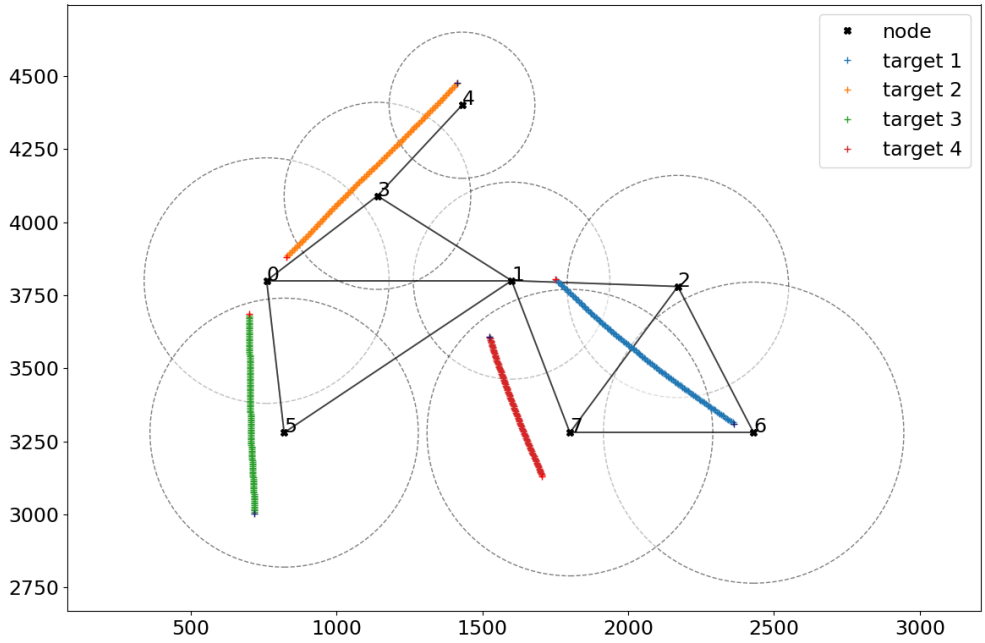


Figure 6.13 Complex scenario

Notable events affecting the performance of this scenario depicted in Figure 6.13 are the following:

- $t = 16$: Target 1 enters the FOV of node 7.
- $t = 25$: Target 2 enters the FOV of node 3. Target 1 enters the FOV of node 2.
- $t = 42$: Target 4 leaves the FOV of node 1.
- $t = 47$: Target 2 leaves the FOV of node 4.
- $t = 78$: Target 1 leaves the FOV of node 7. Target 3 leaves the FOV of node 0.
- $t = 86$: Target 2 enters the FOV of node 0.

- $t = 101$: Target 1 enters the FOV of node 1.
- $t = 130$: Target 2 leaves the FOV of node 3. Target 1 leaves FOV of nodes 6 and 2.

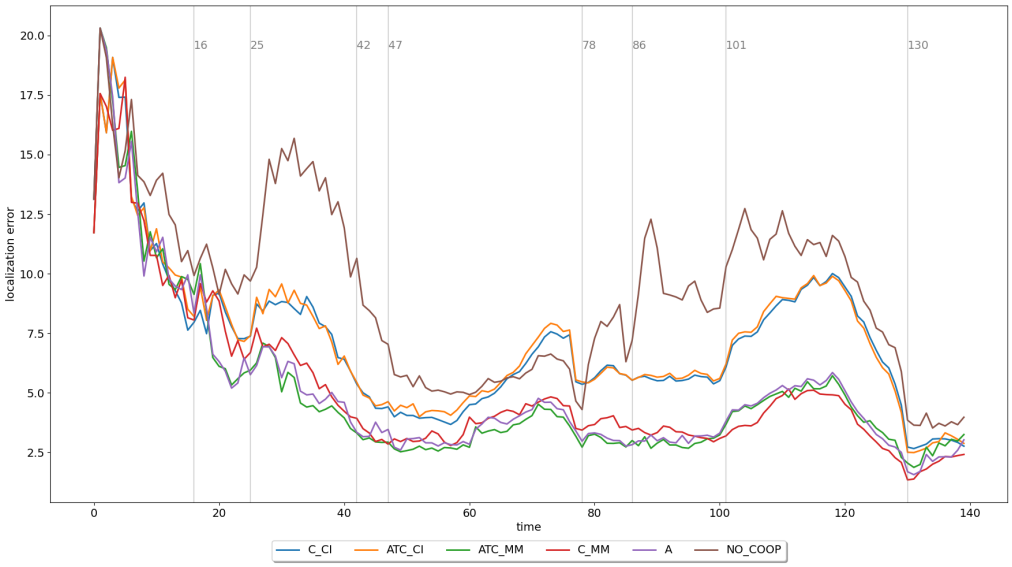


Figure 6.14 Averaged GOSPA for the complex scenario

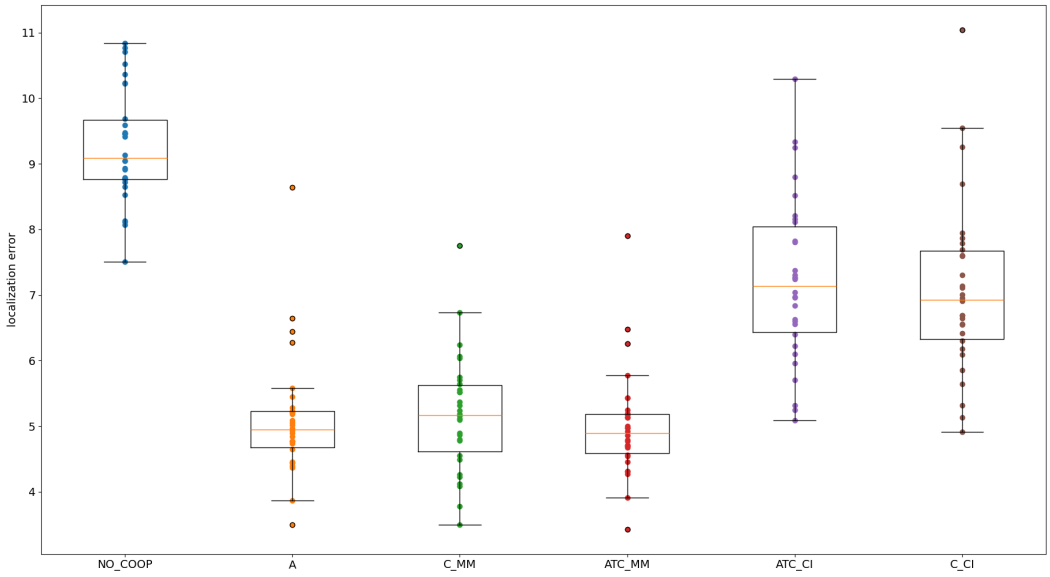


Figure 6.15 Comparison of average localization error over a whole simulation run. One dot represents one simulation run.

In Figures 6.14 and 6.15 we can see that that all methods of cooperation outperform the NO_COOP version.

Conclusion

Main goals of this thesis were:

- Get familiar with the (J)PDA and/or IPDA filters. Study the initialization and deletion of tracks.
- Study the possibilities of collaboration of several filters in a network.
- Propose a method for collaboration among multiple MTT filters.
- Perform an experimental validation of the resulting solution using simulated data.

In Chapter 1 we explained and define necessary mathematical prerequisites. In chapter 2 we introduced the concepts of Hidden markovian model and measurement model. We then followed with derivation of the Kalman filter. We allowed the target to stay undetected sometimes and introduced the concept of false measurements, i.e., clutter, with which the Kalman filter cannot perform satisfyingly well. We then introduced and derived the probablity data association filter as an evolution of the Kalman filter. We also described a performance optimalization known as validation gating, which limits the number of data association, that are needed to be computed. Naturally, targets can appear and disappear at any moment, for this reason we described two possible methods of initializing and deinitializing tracks, the M&N method and IPDA method. Main focus was laid on the latter, which was also described and derived. Next in Chapter 3 we moved from the problem of single target onto the multiple target problem. Even-though, the PDA filter can be used for tracking multiple targets, we introduced another evolution of the filter, namely, the joint probablity data assocation filter. We derived a joint way to compute data assocation coefficients that improves JPDA's performance compared to the PDA filter. Having finished studying concepts of PDA, JPDA and IPDA filters, in Chapter 4 we introduced different general methods of collaboration between filters and focused mostly on the diffusion method. Once we described basic concepts of the diffusion collaboration, in Chapter 5 we propose diffusion

JIPDA filter itself. In the last Chapter 6 we demonstrated performance of the proposed filter.

While the proposed method performs better than its non-collaborative version. It performed poorly in case of two targets moving in similar direction in close proximity.

Besides mitigating the issue mentioned above, the filter could also be further improved by introducing a collaborative way to initialize new target and, potentially, by finding a better method of combining survival probabilities of targets.

Bibliography

[1] B. Wei and B. D. Nener, "Multi-sensor space debris tracking for space situational awareness with labeled random finite sets," *IEEE Access*, vol. 7, pp. 36 991–37 003, 2019. DOI: 10.1109/ACCESS.2019.2904545.

[2] J. L. Tonry, "An early warning system for asteroid impact," *Publications of the Astronomical Society of the Pacific*, vol. 123, no. 899, pp. 58–73, Jan. 2011, ISSN: 1538-3873. DOI: 10.1086/657997.

[3] L. Shuang and C. Pingyuan, "Landmark tracking based autonomous navigation schemes for landing spacecraft on asteroids," *Acta Astronautica*, vol. 62, no. 6, pp. 391–403, 2008, ISSN: 0094-5765. DOI: 10.1016/j.actaastro.2007.11.009.

[4] M. Vetrisano and M. Vasile, "Autonomous navigation of a spacecraft formation in the proximity of an asteroid," *Advances in Space Research*, vol. 57, no. 8, pp. 1783–1804, 2016, Advances in Asteroid and Space Debris Science and Technology - Part 2, ISSN: 0273-1177. DOI: 10.1016/j.asr.2015.07.024.

[5] H. S. Ramos, A. Boukerche, R. W. Pazzi, A. C. Frery, and A. A. Loureiro, "Cooperative target tracking in vehicular sensor networks," *IEEE Wireless Communications*, vol. 19, no. 5, pp. 66–73, 2012. DOI: 10.1109/MWC.2012.6339474.

[6] O. Shorinwa, J. Yu, T. Halsted, A. Koufos, and M. Schwager, "Distributed multi-target tracking for autonomous vehicle fleets," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3495–3501. DOI: 10.1109/ICRA40945.2020.9197241.

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

[8] A. Vasebi, M. Partovibakhsh, and S. M. T. Bathaee, "A novel combined battery model for state-of-charge estimation in lead-acid batteries based on extended kalman filter for hybrid electric vehicle applications," *Jour-*

nal of Power Sources, vol. 174, no. 1, pp. 30–40, 2007, Hybrid Electric Vehicles, ISSN: 0378-7753. DOI: 10.1016/j.jpowsour.2007.04.011.

[9] A. Vasebi, S. Bathaee, and M. Partovibakhsh, "Predicting state of charge of lead-acid batteries for hybrid electric vehicles by extended kalman filter," *Energy Conversion and Management*, vol. 49, no. 1, pp. 75–82, 2008, ISSN: 0196-8904. DOI: 10.1016/j.enconman.2007.05.017.

[10] D. Simon, *Optimal State Estimation: Kalman, H_∞, and Nonlinear Approaches*. Wiley-Interscience, 2006.

[11] X. Rong Li and Y. Bar-Shalom, "Tracking in clutter with nearest neighbor filters: Analysis and performance," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 3, pp. 995–1010, Jul. 1996.

[12] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *IEEE Control Systems Magazine*, vol. 29, no. 6, pp. 82–100, 2009. DOI: 10.1109/MCS.2009.934469.

[13] D. Musicki and R. Evans, "Joint integrated probabilistic data association – JIPDA," in *Proceedings of the Fifth International Conference on Information Fusion*, vol. 2, Annapolis, MD, USA, 2002, pp. 1120–1125.

[14] D. Mušicki, R. Evans, and S. Stankovic, "Integrated probabilistic data association," *IEEE Transactions on Automatic Control*, vol. 39, no. 6, pp. 1237–1241, Jun. 1994.

[15] D. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979.

[16] E. Brekke, *Fundamentals of sensor fusion*. Norwegian University of Science and Technology, 2020.

[17] R. P. S. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Artech House, 2007.

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

[19] B.-T. Vo, B.-N. Vo, and A. Cantoni, "Analytic implementations of the cardinalized probability hypothesis density filter," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3553–3567, Jul. 2007.

[20] R. Mahler, "Phd filters of higher order in target number," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 4, pp. 1523–1543, 2007. DOI: 10.1109/TAES.2007.4441756.

[21] Á. F. García-Fernández and L. Svensson, "Trajectory phd and cphd filters," *IEEE Transactions on Signal Processing*, vol. 67, no. 22, pp. 5702–5714, 2019. DOI: 10.1109/TSP.2019.2943234.

[22] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, 2014. DOI: 10.1109/JPROC.2014.2306253.

[23] B. Chen, L. Yu, W.-A. Zhang, and H. Song, "Distributed fusion kalman filtering with communication constraints," in *2013 American Control Conference*, 2013, pp. 3852–3857. DOI: 10.1109/ACC.2013.6580427.

[24] Ø. K. Helgesen, E. F. Brekke, H. H. Helgesen, and Ø. Engelhardtsen, "Sensor combinations in heterogeneous multi-sensor fusion for maritime target tracking," in *2019 22th International Conference on Information Fusion (FUSION)*, 2019, pp. 1–9. DOI: 10.23919/FUSION43075.2019.9011297.

[25] B. Johansson, T. Keviczky, M. Johansson, and K. Johansson, "Thta12.5 subgradient methods and consensus algorithms for solving convex optimization problems," Jan. 2009, pp. 4185–4190. DOI: 10.1109/CDC.2008.4739339.

[26] K. Dedecius and P. M. Djurić, "Sequential estimation and diffusion of information over networks: A bayesian approach with exponential family of distributions," *IEEE Transactions on Signal Processing*, vol. 65, no. 7, pp. 1795–1809, 2017. DOI: 10.1109/TSP.2016.2641380.

[27] T. Li, M. Mallick, and Q. Pan, "A parallel filtering-communication-based cardinality consensus approach for real-time distributed phd filtering," *IEEE Sensors Journal*, vol. 20, no. 22, pp. 13 824–13 832, 2020. DOI: 10.1109/JSEN.2020.3004068.

[28] M. Üney, D. E. Clark, and S. J. Julier, "Distributed fusion of phd filters via exponential mixture densities," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 521–531, 2013. DOI: 10.1109/JSTSP.2013.2257162.

[29] G. Welch and G. Bishop, "Welch & bishop , an introduction to the kalman filter 2 1 the discrete kalman filter in 1960," 1994. [Online]. Available: https://api.semanticscholar.org/CorpusID:9209711.

[30] P. C. Niedfeldt, "Recursive-ransac: A novel algorithm for tracking multiple targets in clutter," 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:126887335.

[31] M. Sedehi, P. Lombardo, and A. Farina, "A modified m/n logic for track initiation of low observable targets using amplitude information," in *2006 International Radar Symposium*, 2006, pp. 1–4. DOI: 10.1109/IRS.2006.4338080.

[32] D. Musicki, R. Evans, and S. Stankovic, "Integrated probabilistic data association," *IEEE Transactions on Automatic Control*, vol. 39, no. 6, pp. 1237–1241, 1994. DOI: 10.1109/9.293185.

[33] I. Cox and S. Hingorani, "An efficient implementation and evaluation of reid's multiple hypothesis tracking algorithm for visual tracking," in *Proceedings of 12th International Conference on Pattern Recognition*, vol. 1, 1994, 437–442 vol.1. DOI: 10.1109/ICPR.1994.576318.

[34] S. Wang, Q. Bao, and J. Pan, "Multi-bernoulli mixture filter: Complete derivation and sequential monte carlo implementation," in *2021 14th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, 2021, pp. 1–5. DOI: 10.1109/CISP-BMEI53629.2021.9624421.

[35] Á. F. García-Fernández, L. Svensson, J. L. Williams, Y. Xia, and K. Granström, "Trajectory Poisson multi-Bernoulli filters," *IEEE Transactions on Signal Processing*, vol. 68, pp. 4933–4945, 2020, ISSN: 1053-587X, 1941-0476.

[36] M. Beard and S. Arulampalam, "Performance of phd and cphd filtering versus jipda for bearings-only multi-target tracking," in *2012 15th International Conference on Information Fusion*, 2012, pp. 542–549.

[37] D. Smith and S. Singh, "Approaches to multisensor data fusion in target tracking: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 12, pp. 1696–1710, 2006. DOI: 10.1109/TKDE.2006.183.

[38] H. Cho, Y.-W. Seo, B. V. Kumar, and R. R. Rajkumar, "A multi-sensor fusion system for moving object detection and tracking in urban driving environments," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1836–1843. DOI: 10.1109/ICRA.2014.6907100.

[39] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6217–6234, 2012. DOI: 10.1109/TSP.2012.2217338.

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

Enclosed medium contents

- `readme.txt` ..... short description of the media
- `src`
  - `thesis` ..... thesis source code in LaTeX format
  - `impl` ..... implementation source codes
    - `scenarios` ..... generated data for simulations
- `thesis.pdf` ..... text of the thesis in the PDF format