Set JPDA filter for multitarget tracking

Article *in* IEEE Transactions on Signal Processing · November 2011

DOI: 10.1109/TSP.2011.2161294 · Source: IEEE Xplore

CITATIONS
69

READS
665

4 authors, including:

Daniel Svensson
Zenuity AB
26 PUBLICATIONS **303** CITATIONS
SEE PROFILE

Marco Guerriero
GE Global Research
35 PUBLICATIONS **539** CITATIONS
SEE PROFILE

P. Willett
University of Connecticut
667 PUBLICATIONS **11,719** CITATIONS
SEE PROFILE

CHALMERS

Target Tracking in Complex Scenarios

Daniel Svensson

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2010

Thesis for the degree of Doctor of Philosophy

Target Tracking in Complex Scenarios

by

Daniel Svensson

CHALMERS

Department of Signals and Systems
Chalmers University of Technology
Göteborg, Sweden 2010

Target Tracking
in Complex Scenarios
Daniel Svensson
ISBN 978-91-7385-436-8

This thesis has been prepared using LaTeX.

Doktorsavhandlingar vid Chalmers Tekniska Högskola
Ny serie nr 3113
ISSN 0346-718X

Department of Signals and Systems
Signal Processing Group
Chalmers University of Technology
SE-412 96 Göteborg, Sweden

Phone: +46 (0)31 772 1828
Fax: +46 (0)31 772 1748
E-mail: daniel.svensson@chalmers.se

Printed by Chalmers Reproservice
Göteborg, Sweden, September 2010

To my family

Target Tracking in Complex Scenarios

Daniel Svensson
Department of Signals and Systems
Chalmers University of Technology

Abstract

This thesis is concerned with three important components in target tracking, namely multiple-model filtering, data association and sensor resolution modeling. For multiple-model filtering, the preferred method has long been the Interacting Multiple Model (IMM) filter, which relies on the assumption that immediate model shifts have the highest probability. In this thesis, an alternative switching model is proposed, which forces the models to persist for at least a model-specific time, yielding a less complex problem in terms of model hypotheses. Further, a state estimation algorithm is derived, which is close to optimal under the model assumption. The proposed filter, called the Switch-Time Conditioned IMM (STC-IMM) filter, is shown to provide better performance than the IMM filter in benchmark scenarios.

Traditional tracking algorithms are designed to estimate the states of the targets, while trying to maintain their identities. In this thesis, it is shown how these algorithms can be adjusted to problems where target identity is not relevant. More specifically, the Joint Probabilistic Data Association (JPDA) filter is considered, and two adjustments of it are presented, called the Set JPDA (SJPDA) and the Kullback-Leibler Set JPDA (KLSJPDA) filters. These filters both enable more accurate Gaussian approximations, and provide more accurate state estimates than the JPDA filter when evaluated with a metric that disregards identity. Another approach to the problem is to use Finite Set Statistics (FISST). In the thesis, the results of the first performance comparison of the most prominent FISST-based and traditional filters are presented and discussed.

In the development of most tracking algorithms, it is assumed that the targets are always resolved by the sensor. However, when the targets are closely spaced in relation to the sensor resolution, this assumption is not valid, and may lead to decreased tracking performance. This thesis presents a multi-target sensor resolution model, for an arbitrary but known number of targets, which takes resolution effects into account. It is further shown how the model is incorporated into a Bayesian tracking framework, and two alternative JPDA-like filters are presented.

Keywords: Target tracking, state estimation, multiple-model filtering, random finite sets, performance evaluation, sensor models, radar.

Sammanfattning

I den här avhandlingen behandlas tre viktiga delar av ett målföljningssystem, nämligen filtrering med multipla modeller, dataassociation, och modellering av sensorupplösning. Den vanligaste metoden att hantera multipla modeller är att använda det så kallade IMM-fitret, där IMM står för "Interacting multiple model". IMM-filtret baseras på antagandet att sannolikheten är störst att byten mellan modeller sker omedelbart. I den här avhandlingen föreslås istället ett alternativt modellbyteskriterie, vilket tvingar en modell att vara aktiv en viss minsta modellspecifika tid. Genom den här modellen fås ett mindre komplext problem sett till antalet modellhypoteser, och för det problemet härleds ett följefilter, vilket är nästan optimalt för fall då modellen stämmer väl överens med verkligheten.

Traditionella följefilter har tagits fram för att hantera följeproblem där målidentitet är av vikt. Även om det i många fall är viktigt med mål-ID, så finns det också ett flertal applikationer där det inte är relevant. I avhandlingen visas det hur de traditionella algoritmerna kan anpassas för följeproblem där mål-ID inte är av intresse. Mer specifikt studeras JPDA-filtret, och två justeringar av filtret, som vi kallar "Kullback-Leibler Set JPDA" (KLSJPDA) och "Set JPDA" (SJPDA), presenteras. Med båda dessa filter möjliggörs noggrannare Gaussapproximationer och mer precisa skattningar, vilket visas av utvärderingar på simulerade data. För justeringen utnyttjas sambandet mellan traditionella metoder och "Finite set statistics" (FISST), som är en alternativ metod att hantera följning utan målidentitet, vilken rymmer ett flertal filter. I avhandlingen presenteras vidare en utvärderingsstudie av prestandan hos de två mest sofistikerade följefiltren inom respektive familj av algoritmer. För utvärderingen används ett prestandamått, "Mean Optimal Subpattern Assignment" (MOSPA), som inte tar hänsyn till målidentitet. Detta mått ligger även till grund för utvecklingen och analysen av de tidigare nämnda filtren SJPDA och KLSJPDA.

I de allra flesta fall då följefilter utvecklas och används så antas det att alla mål alltid är upplösta. För fall där mål befinner sig nära varandra i förhållande till sensorupplösningen så är detta ett antagande som stämmer dåligt överens med verkligheten, och som kan leda till dåliga prestanda. För att råda bot på detta presenteras i den här avhandlingen en upplösningsmodell för multipla mål (av känt antal), vilken tar hänsyn till upplösningsproblematik. Vi beskriver även hur modellen kan användas i ett Bayesianskt följeramverk, samt visar hur två JPDA-filter kan tas fram baserat på detta.

Nyckelord: Målföljning, tillståndsskattning, filtrering med multipla modeller, random finite sets, prestandautvärdering, sensormodeller, radar.

Preface

The path to the writing of a PhD thesis is not straight. Nor is it broad, well-cut, and without danger. However, when the end of it is near, the burdens are forgotten, the struggle is over, and you can look forward into the light!

By the above passage, I do not want to shed a dark and negative light on my journey to the finalization of this thesis; on the contrary, rather, it has been delightful four and a half years, which have been developing both professionally and personally. This is most certainly due to the excellent research environment in the Signal Processing Group at Chalmers, where a balance between work and social life is considered important; and due to the current and former colleagues in the group who are both skilled researchers and fun people.

I would like to end by expressing my gratitude to all those people who, in one way or another, have been part of this journey, and who have made it easier and more enjoyable—at work and outside of work.

Daniel Svensson

Göteborg, September 8, 2010

Acknowledgement

I would like to start by acknowledging Prof. Mats Viberg for letting me join his research group, which throughout my time in the group have been filled with talented and fun people—many of which have shared my interests outside of work.

My second thank you goes to my supervisor Lennart Svensson, for his guidance and support throughout my PhD studies. While starting off as a pure supervisor–student relation, we have gradually transformed our work relation into that of a colleagueship and co-workership, which I appreciate. Also, I would like to thank Lennart for nice and fruitful discussions, which quite often have taken place at the tennis court (which I hope both of us have enjoyed, despite my fierce slices).

Most of the research presented in this thesis is the result of good collaborations with talented people. I feel lucky to have had the possibility of working with Lennart Svensson, Johannes Wintenby, Marco Guerriero, Peter Willett, Martin Ulmke, and Lars Danielsson. To Martin Ulmke I am especially grateful for inviting me to the former FGAN institute in Wachtberg, Germany, and for all his support during my two visits at the institute—ranging from asking his neighbors if they could lend me their car for two months, to late evening debugging of code. After several Rossö workshops, I am also glad that I got the opportunity to collaborate with Lars Danielsson during my final year.

The main parts of my research have been conducted within two consecutive research projects in which VINNOVA and Electronic Defence Systems, Saab AB have been the main sponsors. To my colleagues and co-supervisors at Saab—Åke Andersson and Johannes Wintenby—I would like to say a big thank you for many rewarding discussions, and for help and support.

For proofreading the thesis, I am indebted to Malin Lundgren, Åke Andersson, Lennart Svensson, and Mats Viberg. The thesis indeed became a

better one thanks to your comments.

Writing a thesis is so much easier when you have fun on your spare time. For contributing to this, I am very grateful to my brothers Mikael and Thomas; and to Peter, Emma, Frida, Daniel and Sofia. To Peter I send an extra thank you for giving me rides to and from work during this last year. Finally, I would like to express my gratitude to my parents, who are always there for me, no matter what.

List of Publications

This thesis is based on the following publications:

Paper I
D. Svensson and L. Svensson. A New Multiple Model Filter with Switch Time Conditions. In *IEEE Transactions on Signal Processing*, vol. 58, no. 1, January 2010.

Paper II
L. Svensson, D. Svensson, M. Guerriero and P. Willett. Set JPDA Algorithm for Multi-Target Tracking. Submitted to *IEEE Transactions on Signal Processing*.

Paper III
D. Svensson, J. Wintenby and L. Svensson. Performance Evaluation of MHT and CPHD in a Ground Target Tracking Scenario. In *Proceedings of the 12th International Conference on Information Fusion*, Seattle, USA, 2009.

Paper IV
D. Svensson, M. Ulmke and L. Danielsson. Multitarget Sensor Resolution Model and Joint Probabilistic Data Association. Submitted to *IEEE Transactions on Aerospace and Electronic Systems*.

Other publications by this author, but omitted in the thesis:

- L. Svensson and D. Svensson. Multiple Model Filtering with Switch Time Conditions. In *Proceedings of the 10th International Conference on Information Fusion*, Qubec, Canada, 2007.

- D. Svensson and L. Svensson. An Alternative Derivation of the Gaussian Mixture Cardinalized Probability Hypothesis Density Filter. Tech. report no. R018/2008, Department of Signals and Systems, Chalmers University of Technology, Göteborg, Sweden, November 2008.

- L. Svensson, D. Svensson and P. Willett. Set JPDA algorithm for tracking unordered sets of targets. In *Proceedings of the 12th International Conference on Information Fusion*, Seattle, USA, 2009.

- D. Svensson, M. Ulmke and L. Danielsson. Multitarget sensor resolution model for arbitrary target numbers. In *Signal and Data Processing of Small Targets*, ser. Proc. SPIE, vol. 7698, Orlando, USA, 2010.

- D. Svensson, M. Ulmke and L. Danielsson. Joint probabilistic data association filter for partially unresolved target groups. In *Proceedings of the 13th International Conference on Information Fusion*, Edinburgh, UK, 2010.

- M. Guerriero, L. Svensson, D. Svensson and P. Willett. Shooting two birds with two bullets: how to find minimum mean OSPA estimates. In *Proceedings of the 13th International Conference on Information Fusion*, Edinburgh, UK, 2010. Winner of the Jean-Pierre Le Cadre Best Paper Award.

- L. Svensson and D. Svensson. Set JPDA Filter for Multi-Target Tracking. Tech. report no. R012/2010, Department of Signals and Systems, Chalmers University of Technology, Göteborg, Sweden, September 2010.

- D. Svensson, M. Ulmke and L. Danielsson. Multi-Target Tracking with Partially Unresolved Measurements. In *Proceedings of the IEEE ISIF Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF 2010)*, Leipzig, Germany, 2010.

Contents

Part I
Introduction

Chapter 1

Introduction

Target tracking is an ever-increasing field of research with a wide spectrum of applications. In the early dawn of the research area, the main driving force was military applications, such as the detection and estimation of the positions, speeds and directions of incoming aircraft, using ground-based radar systems. However, as the field has developed and matured, the methods have spread to disparate domains, such as bio-medicine, finance, automotive safety, and air traffic control. The foundation of tracking is to recursively estimate an unknown quantity over time, be it the temperature in a room, the value of a share on the stock market, the positions of aircraft circling around an airport, or the movement of a cell in a blood vessel. In the traditional setting, the quantities of interest are the positions and speeds of objects (called targets), which are often referred to as states.

A first step to perform tracking is to make a prediction of the future value of the state. To do so, a model that describes the evolution of the quantities of interest is required. Such a model is referred to as a *process model*, or, in the case of target tracking, a *motion model*. Through the motion model, we can predict the future value of the state, given knowledge of its current value. We can also express the uncertainties in that prediction. For example, if the problem is to track a car on a road, a model for its dynamics can be that it moves with slowly varying speed along the road. By knowing the current position, speed and direction of the car, the motion model can be used to predict its future position. Of course, the longer the prediction horizon, the more uncertain the prediction will be, and this should also be reflected in the motion model. An example of this is seen in weather forecasting, where the uncertainties in the predictions increase with the number of days; the prediction for the coming day is rather certain, while a prediction five days ahead is very uncertain.

When the time has been reached to which a prediction was made, we would like to update the predicted value using additional information. In a tracking system, such a source of information is some type of sensor—traditionally a radar, but other popular sensors are cameras and laser sensors. To benefit from the information provided by one or several sensors, the relation between the quantity of interest and the output of the sensors must be modeled, i.e., we need to define a *sensor model*, often called a *measurement model*. In the simplest case, the measurement is equal to the true value of the quantity, while a more realistic model is to model the measurement as the true value plus noise, where the noise component is introduced to incorporate errors in the model as well as potential impreciseness of the sensor. Apart from the accuracy and model uncertainties, it can also be the case that all targets are not always detectable, or that they are so closely spaced that they appear as one object . For accurate modeling of the sensor measurements, such aspects must also be considered and accounted for.

The title of this thesis is "Target Tracking in Complex Scenarios". In the following, we discuss different examples of complex scenarios. We also discuss in what way this thesis contributes to some of these research problems. In the ideal tracking scenario, the targets are widely separated, they are always detectable by the sensor, and there are no false, or spurious, detections (due to, e.g., receiver noise, or unwanted signal returns from the ground, or the surrounding environment, called clutter). For such scenarios, the solution to the tracking problem is straightforward, and readily given by standard methods. However, in many practical systems, this ideal situation rarely occurs. Instead, the targets are not always detected, false alarms are reported, and the targets are intermittently closely spaced, or even occluded. Further, unwanted signal returns from uninteresting objects (for example wind power mills) may be received, and intentional jamming signals could be transmitted by an adversary. Then, the scenario is more complex, and the tracking algorithms must rely on more sophisticated methods. It can also be so that the targets themselves are complex: either by having a complex structure such that the signal returns from them are hard to model, or by having dynamics which are difficult to model, or which are not accurately described by a single model. The research projects within which the work supporting this thesis has been carried out consider target tracking in complex scenarios using airborne radar. The problem is then to detect and track targets on the ground, using measurements from radar systems mounted on aircraft—a problem referred to as *ground target tracking*.

The process of predicting the future states of the targets, and to update those predictions using sensor measurements, is called *filtering*. In many applications, the prediction can be performed using a single motion model.

Often, a simple model is used; for example a model that describes the speed as constant, with small perturbations. However, if the target has two distinctively different modes of operation, which is swaps between, a description of the movement by a single model might not yield the accuracy in the estimation that the user requires. For example, an aircraft normally flies either straight ahead, or performs a maneuver, and a vehicle could either travel on-road or off-road. A better approach is then to model the dynamics by two motion models. The incorporation of several motion or measurement models into the filtering is referred to as *multiple model filtering*. In Paper I, a new model structure for multiple model filtering is proposed, which is claimed to be a better description of the behavior of many targets. A multiple model filter based on that model structure is also presented.

To account for information provided by a sensor, the relation between the target states and the measurements must be described. Often, an assignment procedure is carried out, where the measurements are classified as target-generated or false, and where an association of target-generated measurements to targets is performed. This procedure is referred to as *data association*. When targets are widely separated, the problem is straightforward, but when the targets are closely spaced, it becomes difficult. Also, when the detection probability is lower than one, and when the sensor reports false measurements, the data association problem requires more advanced algorithms. Within ground target tracking, this occurs, e.g., when a convoy of targets are tracked. In Paper III, two prominent, and conceptually different, data association algorithms are evaluated on a ground target tracking scenario, and conclusions are drawn regarding their behavior.

Another issue of closely-spaced targets is sensor resolution, where the resolution describes the ability of a sensor to discriminate between targets that are near each other. When designing a sensor system, such as a radar system, the resolution capability is often chosen to match the size of the objects of interest, in order to receive as much signal energy as possible from each target. However, if the sizes of the targets are different, resolution problems can occur. For example, targets that are smaller than the resolution cell can become unresolvable by the sensor, and only provide a single joint measurement, while targets that are larger than the resolution cell can span over several cells and hence yield several detections. These problems are common in ground target tracking, since vehicles on the ground (car, trucks, boats and trains) have widely different sizes. In Paper IV, the problem of limited sensor resolution is considered, and a modeling framework for considering resolution limitations in a tracking filter is proposed. Among the subjects considered in this thesis, the resolution modeling is the one which has received the least attention in the literature, although it is generally

accepted as an important problem.

In target tracking, there are in principle two philosophically different directions. In the traditional approach, the problem is to estimate where each target is, whereas another approach is to consider the problem of estimating where there are targets, disregarding their identities (labels). For example, the user could be interested in determining in which areas there are objects, but not be interested in which one is which, and where each specific target was previously. In collision avoidance systems, for example, there is no interest in which car (or boat) is which, the only interest is in avoiding all vehicles. In Paper II, it is shown that there is a link between the two conceptually different approaches, and it is shown how the methods developed for the tracking problem with target identity can be adjusted to the label-free problem.

1.1 Research Project and Support

The research that has led up to the writing of this thesis has mainly been performed within two consecutive research projects which are part of the Swedish research program "Nationella flygtekniska forskningsprogrammet" (NFFP)—a program financed by Vinnova (The Swedish Governmental Agency for Innovation Systems). The first project, called "Ground target tracking using airborne radar systems", was part of the program NFFP4, and it was a joint project between the Signal Processing Group at Chalmers University of Technology, and Electronic Defence Systems, Saab AB. The second project, called "Target tracking in complex scenarios for airborne radar systems", is part of the ongoing program NFFP5. This project is a collaboration between the two aforementioned parties, and Saab Bofors Dynamics AB.

The material presented in Paper IV is the result of a research visit by the author of this thesis to the Fraunhofer-FKIE institute in Wachtberg, Germany. For additional support of that visit, scholarships were granted by the German Academic Exchange Service, and Bernt Järmarks stiftelse.

1.2 Outline of the Thesis

This thesis is divided into two parts: in the first part, the theoretical background of the thesis work is presented, with the purpose of introducing the topic and of preparing the reader for its second part. In that second part, the contributions of the author to the field of target tracking and sensor modeling are presented in the form of four appended papers.

The first part of the thesis is structured as follows. In Chapter 2, the

concepts of single and multiple model filtering are introduced, and two commonly used filters are presented. Chapter 3 considers data association under two conceptually different tracking settings, namely tracking with or without interest in the identity of the targets. For each setting, the most commonly used algorithms are presented and discussed. In Chapter 4, radar measurement modeling is considered. In particular, the general measurement principle, and the notions of accuracy and resolution are described. Finally, in Chapter 5, the contributions of this thesis are presented and summarized, and future work and directions within the area of target tracking are discussed.

Chapter 2

Multiple Model Filtering

A tracking algorithm generally consists of at least two parts: filtering and data association. In this chapter, the filtering part is considered, while the next chapter is devoted to data association and, to some extent, track handling. By using data collected over the course of time, filtering is about recursively estimating an unknown quantity, be it the temperature in a room, the value of a share on the stock market, or the position of a vehicle on a road. The filtering is made up of two steps: *prediction* and *measurement update*. In the prediction step, the future state of the unknown quantity is predicted, given knowledge about the current state, and in the update step, measurements are utilized to improve upon that prediction.

In target tracking, the unknown quantities are the target states, where a state is, e.g., the position and speed of an object. The task of a filter is then to estimate those states, using information from the received measurements. To simplify the description, we here present filtering for a single target. The extension to multiple targets, however, is straightforward if the number of targets, and how they relate to the received measurements, are known.

2.1 Single Model Filtering

We start by considering the single-model filtering problem. First the problem formulation is stated, then the conceptual solution to the problem is discussed, and finally the Kalman filter is introduced, which is the optimal single-model filter under certain assumptions.

2.1.1 Problem Formulation

The unknown state of the target under consideration is denoted $\mathbf{x}_k$, where sub-index k indicates the current discrete time index. Most real-world processes are continuous in time, where the continuous time instant t_k corresponds to discrete time index k. The measurements of a sensor, however, are almost always produced at certain discrete moments in time. Although the filtering can be performed in continuous time, it is most commonly done in discrete time, which is the description used in this chapter.

At a certain time t_k, we would like to estimate $\mathbf{x}_k$ with as high accuracy as possible. We thus seek an estimate, $\hat{\mathbf{x}}_{k|k}$, which is close to the true state in some sense. In a Bayesian setting [1], the estimate is derived from the posterior probability density function (pdf)

$$p\big(\mathbf{x}_k \big| \mathbf{Z}^k\big),$$

where $\mathbf{Z}^k$ represents a sequence of measurements from time 1 to time k. A measurement need not be a scalar value, but can be a vector of measures, e.g., positions, velocities and accelerations in several dimensions. The sequence of measurements is then an ordered set

$$\mathbf{Z}^k = \{\mathbf{z}_1, \ldots, \mathbf{z}_k\} \tag{2.1}$$

of measurement vectors $\mathbf{z}_k$.

If the posterior pdf is known, we can extract estimates of the target state. A popular estimate is the minimum mean-square error (MMSE) estimate, which is given by

$$\hat{\mathbf{x}}_k^{\text{MMSE}} = \mathbb{E}\left\{\mathbf{x}_k \big| \mathbf{Z}^k\right\} = \int \mathbf{x}_k \, p\big(\mathbf{x}_k \big| \mathbf{Z}^k\big) d\mathbf{x}_k, \tag{2.2}$$

where $\mathbb{E}$ denotes expectation. The objective of filtering is to calculate the posterior pdf $p\big(\mathbf{x}_k \big| \mathbf{Z}^k\big)$, or an approximation of it, and to extract an estimate, $\hat{\mathbf{x}}_{k|k}$, from that density.

2.1.2 Conceptual Solution

As we discussed in the previous section, filtering for a single target includes the exact or approximate calculation of the posterior pdf $p\big(\mathbf{x}_k \big| \mathbf{Z}^k\big)$. In this section, we first describe the exact calculation of the density, and then present the Kalman filter, which under certain assumptions provides a closed-form solution to the single-model filtering problem.

We start by splitting the measurement set into two parts, giving

$$p\big(\mathbf{x}_k\big|\mathbf{Z}^k\big) = p\big(\mathbf{x}_k\big|\mathbf{z}_k, \mathbf{Z}^{k-1}\big). \tag{2.3}$$

Bayes' rule [1] is then used to rewrite the posterior pdf as

$$p\big(\mathbf{x}_k\big|\mathbf{Z}^k\big) = \frac{p\big(\mathbf{z}_k\big|\mathbf{x}_k, \mathbf{Z}^{k-1}\big)p\big(\mathbf{x}_k|\mathbf{Z}^{k-1}\big)}{p\big(\mathbf{z}_k\big|\mathbf{Z}^{k-1}\big)}. \tag{2.4}$$

Given the target state $\mathbf{x}_k$, the density of the measurement vector does not depend on previous measurements, so

$$p\big(\mathbf{x}_k\big|\mathbf{Z}^k\big) = \frac{p\big(\mathbf{z}_k\big|\mathbf{x}_k\big)p\big(\mathbf{x}_k|\mathbf{Z}^{k-1}\big)}{p\big(\mathbf{z}_k\big|\mathbf{Z}^{k-1}\big)}. \tag{2.5}$$

A filter that calculates the posterior density of a single target according to equation (2.5) is called a *single-target Bayes filter*.

As given by (2.5), the posterior density is proportional to the product of a likelihood, $p\big(\mathbf{z}_k\big|\mathbf{x}_k\big)$, and a prior, $p\big(\mathbf{x}_k|\mathbf{Z}^{k-1}\big)$. The likelihood describes how likely the state vector $\mathbf{x}_k$ is, given that we have made the observation $\mathbf{z}_k$, and is often thought of as a function of $\mathbf{x}_k$. In target tracking, $p\big(\mathbf{z}_k\big|\mathbf{x}_k\big)$ is referred to as the *measurement model*, or sensor model. Further, the prior is found by marginalizing over the previous state, $\mathbf{x}_{k-1}$,

$$\begin{aligned} p\big(\mathbf{x}_k\big|\mathbf{Z}^{k-1}\big) &= \int p\big(\mathbf{x}_k, \mathbf{x}_{k-1}\big|\mathbf{Z}^{k-1}\big)d\mathbf{x}_{k-1} \\ &= \int p\big(\mathbf{x}_k\big|\mathbf{x}_{k-1}, \mathbf{Z}^{k-1}\big)p\big(\mathbf{x}_{k-1}\big|\mathbf{Z}^{k-1}\big)d\mathbf{x}_{k-1}. \end{aligned} \tag{2.6}$$

Assuming that the target dynamics fulfill the Markov property [2], i.e.,

$$p\big(\mathbf{x}_k\big|\mathbf{x}_{k-1}, \mathbf{Z}^{k-1}\big) = p\big(\mathbf{x}_k\big|\mathbf{x}_{k-1}\big), \tag{2.7}$$

we get the Chapman-Kolmogorov equation

$$p\big(\mathbf{x}_k\big|\mathbf{Z}^{k-1}\big) = \int p\big(\mathbf{x}_k\big|\mathbf{x}_{k-1}\big)p\big(\mathbf{x}_{k-1}\big|\mathbf{Z}^{k-1}\big)d\mathbf{x}_{k-1}. \tag{2.8}$$

The integral in (2.8) propagates the posterior density $p\big(\mathbf{x}_{k-1}\big|\mathbf{Z}^{k-1}\big)$ at time $k-1$ through the pdf $p\big(\mathbf{x}_k\big|\mathbf{x}_{k-1}\big)$, which is called the *motion model* or process model. The resulting density, $p\big(\mathbf{x}_k\big|\mathbf{Z}^{k-1}\big)$, is often referred to as the predicted density.

Using Equations (2.5) and (2.8), we have a way of expressing the posterior density at time k, $p\big(\mathbf{x}_k\big|\mathbf{Z}^k\big)$, as a function of the previous posterior,

$p\big(\mathbf{x}_{k-1}\big|\mathbf{Z}^{k-1}\big)$, the motion model, $p\big(\mathbf{x}_k\big|\mathbf{x}_{k-1}\big)$, and the measurement model, $p\big(\mathbf{z}_k\big|\mathbf{x}_k\big)$. If the filtering process is initiated with a prior density $p\big(\mathbf{x}_0\big)$ on the initial target state $\mathbf{x}_0$, the equations can be used to first calculate $p\big(\mathbf{x}_1\big|\mathbf{Z}^1\big)$, then $p\big(\mathbf{x}_2\big|\mathbf{Z}^2\big)$, and so on. We thus have a *recursive* way of calculating the posterior pdf whenever a new measurement is received.

An alternative way of describing the motion and measurement models is the system of equations

$$\mathbf{x}_k = f_{k-1}\big(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}\big) \tag{2.9}$$

$$\mathbf{z}_k = h_k\big(\mathbf{x}_k, \mathbf{w}_k\big), \tag{2.10}$$

where f_{k-1} and h_k are possibly non-linear functions, $\mathbf{v}_{k-1}$ is a process noise realization and $\mathbf{w}_k$ is a measurement noise realization. Knowing the motion and measurement models is then equivalent to knowing the functions f_{k-1}, h_k and the joint density functions of $\mathbf{v}_{k-1}$ and $\mathbf{w}_k$. We next use this formulation to approach the problem.

The Kalman Filter

If both the motion model in (2.9) and measurement model in (2.10) are linear, with additive Gaussian noise, it can be shown that the posterior pdf is also Gaussian, provided that the prior $p\big(\mathbf{x}_0\big)$ is Gaussian. A closed-form solution to the filtering problem is then possible, and that solution is given by the *Kalman filter* [3]. For linear models, the Kalman filter recursively calculates the first two moments of the posterior state vector $\mathbf{x}_k\big|\mathbf{Z}^k$, *viz.* the mean and the covariance matrix. Since the first two moments completely characterize a Gaussian pdf, the output of the Kalman filter gives a complete description of $p\big(\mathbf{x}_k\big|\mathbf{Z}^k\big)$. Also, if we as estimator use that mean value, the Kalman filter is the optimal estimator in the mean square error (MSE) sense, since the posterior mean is the MMSE estimator (cf. (2.2)). If the models are linear, but non-Gaussian, the Kalman filter is still the linear MMSE estimator, and it is hence often used also for such models.

In the linear-Gaussian case, the motion and measurement models are described by (cf. (2.9)–(2.10))

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{v}_{k-1} \tag{2.11}$$

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{w}_k, \tag{2.12}$$

where $\mathbf{v}_{k-1}$ and $\mathbf{w}_k$ are independently distributed as

$$\mathbf{v}_{k-1} \sim \mathcal{N}\big(\mathbf{0}, \mathbf{Q}_{k-1}\big) \tag{2.13}$$

$$\mathbf{w}_k \sim \mathcal{N}\big(\mathbf{0}, \mathbf{R}_k\big), \tag{2.14}$$

in which $\mathbf{Q}_{k-1}$ is the process noise covariance matrix, and $\mathbf{R}_k$ is the measurement noise covariance matrix. Furthermore, $\mathbf{F}_{k-1}$ is called the system matrix, and $\mathbf{H}_k$ the observation matrix.

The Kalman filter operates recursively with a prediction step and a measurement update step, in the same manner as the single-target Bayes filter in (2.5). In the prediction step, the Kalman filter produces an estimate, $\hat{\mathbf{x}}_{k|k-1}$, of $\mathbf{x}_k$ using data up to $k-1$. It also describes the accuracy of that estimate through the covariance matrix $\mathbf{P}_{k|k-1}$. In the measurement update step, the prediction estimate $\hat{\mathbf{x}}_{k|k-1}$ is corrected using information from the measurement $\mathbf{z}_k$, yielding the posterior estimate $\hat{\mathbf{x}}_{k|k}$, and covariance matrix $\mathbf{P}_{k|k}$. The Kalman filter represents the posterior pdf as $p\big(\mathbf{x}_k\big|\mathbf{Z}^k\big) \cong \mathcal{N}\big(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}\big)$, where the representation is exact for the linear-Gaussian case.

The prediction step of the Kalman filter is governed by the following equations:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1|k-1} \tag{2.15}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}. \tag{2.16}$$

Further, the update step is described by

$$\tilde{\mathbf{z}}_k = \mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1} \tag{2.17}$$

$$\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k \tag{2.18}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{S}_k^{-1} \tag{2.19}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\tilde{\mathbf{z}}_k \tag{2.20}$$

$$\mathbf{P}_{k|k} = \big(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\big)\mathbf{P}_{k|k-1}. \tag{2.21}$$

The vector $\tilde{\mathbf{z}}_k$ is called the innovation, and it denotes the difference between the received measurement $\mathbf{z}_k$ and the predicted measurement $\mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}$. The matrix $\mathbf{S}_k$ is the covariance matrix of the innovation, hence called the innovation covariance matrix. Finally, $\mathbf{K}_k$ is the Kalman gain, and it is the optimal weighting matrix of the information from the prediction step and the new measurement.

The Kalman filter can only operate on linear models. To handle nonlinear models, two extensions to the Kalman filter have been proposed, called the Extended Kalman Filter (EKF) [4] and the Unscented Kalman Filter (UKF) [5]. Other filtering methods, possible to use for general models, are particle filters [6–8], and grid-based methods [8]. With these approaches the probability density function is approximated, which is different to the EKF, for example, in which the state space model is approximated.

2.2 Filtering with Multiple Models

In some situations, the motion of a target cannot be accurately captured by a single model. Consider for example tracking of an aircraft which typically flies either straight ahead or performs a maneuver. Since the aircraft flies at high speed, the difference in position of the aircraft if it flies straight ahead, compared to if it performs a maneuver, is large. Moreover, if the aircraft has been in the same type of movement for some time, the movement will probably remain for a while longer. It would then be beneficial to describe the motion by two motion models, where a well performing filter mainly relies on the model that describes the current motion the best. Correspondingly, there are similar scenarios where the measurements are better described by multiple measurement models. In situations where there is a dependence over time regarding the nature of the motion or the measurements, a better description that uses multiple models would enable more accurate modeling and thus filtering. This is the foundation of *multiple model filtering*.

A general multiple-model filtering setting is governed by the hybrid system

$$\mathbf{x}_k = f_{k-1}\big(\mathbf{x}_{k-1}, m_k, \mathbf{v}_{k-1}\big) \tag{2.22}$$
$$\mathbf{z}_k = h_k\big(\mathbf{x}_k, m_k, \mathbf{w}_k\big). \tag{2.23}$$

In (2.22)–(2.23), the variable m_k describes which motion model and which measurement model that is active in the time interval $\big(k-1,\, k\big]$. The total number of models is M. The assumption is hence that only one of the models can be active during each time interval. The model variable, m_k, is also called the regime variable. To complete the hybrid system, a model that describes the transitions of the regime variable is required. A common choice is a first-order Markov chain model [9], as described in the next section.

2.2.1 Markov-Based Model Switching

A common model for the regime variable m_k in the hybrid system (2.22)–(2.23) is an M-state first-order Markov chain, with transition probabilities

$$\pi_{ij,k} = \Pr\{m_k = j \big| m_{k-1} = i\}, \quad \big(i, j\big) \in \{1, 2, \ldots, M\}\,. \tag{2.24}$$

The hybrid system is then referred to as a *Jump Markov System*. A homogeneous chain is often assumed, for which $\pi_{ij,k} = \pi_{ij}$.

An important family of jump Markov systems is the jump Markov *linear*

system (JMLS), described by

$$\mathbf{x}_k = \mathbf{F}_{k-1}(m_k)\mathbf{x}_{k-1} + \mathbf{v}_{k-1}(m_k) \tag{2.25}$$
$$\mathbf{z}_k = \mathbf{H}_k(m_k)\mathbf{x}_k + \mathbf{w}_k(m_k). \tag{2.26}$$

Even though the models are linear, the overall system is non-linear unless we know the active model m_k.

To describe the posterior distribution $p(\mathbf{x}_k|\mathbf{Z}^k)$, we marginalize over all possible sequences of models, or regimes. A sequence of regimes can be represented as a path through a model hypothesis tree; see Fig. 2.1 for a two-model example. We first define the regime history

$$\mathcal{M}_k^l = \{m_1^l, \ldots, m_k^l\}, \quad l = 1, \ldots, M^k, \tag{2.27}$$

where m_k^l is the active model between $k-1$ and k, for the l^{th} path through the hypothesis tree. At each time step, each branch in the hypothesis tree is split into M branches, representing a transition to each of the M models. Therefore, the number of possible paths through the tree at time k is M^k. Conditioning on $\mathcal{M}_k^l$, the posterior density $p(\mathbf{x}_k|\mathcal{M}_k^l, \mathbf{Z}^k)$ of $\mathbf{x}_k$ is Gaussian, and is calculated by a Kalman filter. The posterior density is hence given by the Gaussian mixture

$$p(\mathbf{x}_k|\mathbf{Z}^k) = \sum_{l=1}^{M^k} p(\mathbf{x}_k|\mathcal{M}_k^l, \mathbf{Z}^k) \Pr\{\mathcal{M}_k^l|\mathbf{Z}^k\}. \tag{2.28}$$

Further, the regime sequence probability

$$\begin{aligned}
\Pr\{\mathcal{M}_k^l|\mathbf{Z}^k\} &= \Pr\{\mathcal{M}_k^l|\mathbf{z}_k, \mathbf{Z}^{k-1}\} \\
&\propto p(\mathbf{z}_k|\mathcal{M}_k^l, \mathbf{Z}^{k-1}) \Pr\{\mathcal{M}_k^l|\mathbf{Z}^{k-1}\} \\
&= p(\mathbf{z}_k|\mathcal{M}_k^l, \mathbf{Z}^{k-1}) \Pr\{m_k, \mathcal{M}_{k-1}^l|\mathbf{Z}^{k-1}\} \\
&= p(\mathbf{z}_k|\mathcal{M}_k^l, \mathbf{Z}^{k-1}) \Pr\{m_k^l|\mathcal{M}_{k-1}^l, \mathbf{Z}^{k-1}\} \\
&\qquad \times \Pr\{\mathcal{M}_{k-1}^l|\mathbf{Z}^{k-1}\}.
\end{aligned} \tag{2.29}$$

Under the Markovian assumption,

$$\Pr\{m_k^l|\mathcal{M}_{k-1}^l, \mathbf{Z}^{k-1}\} = \Pr\{m_k^l|m_{k-1}^l\} = \pi_{m_{k-1}^l m_k^l} \tag{2.30}$$

is the transition probability between model m_{k-1}^l and m_k^l, assuming that the transition probabilities are independent of the target state $\mathbf{x}_{k-1}$. The conceptual filtering solution to JMLSs, as presented here, is not feasible in practice, since the number of regime histories grow exponentially with time. We thus need to consider sub-optimal solutions.

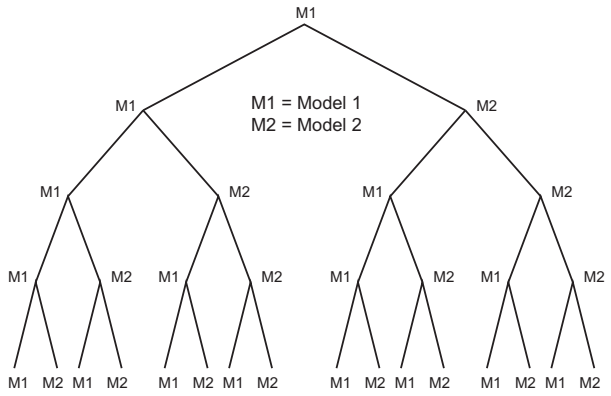

Figure 2.1: Model hypothesis tree for two-model jump-Markov systems. In the figure, it is assumed that model 1 is valid at the first time instant.

Two possible approaches for reducing the number of regime sequences are pruning and merging, where pruning resembles removal of low-probability regime sequences, and merging is equivalent to the unification of several regime sequences into one. A sub-optimal pruning algorithm for JMLS is the Multiple Model Pruning (MMP) algorithm [10], while a popular sub-optimal merging algorithm is the Interacting Multiple Model (IMM) filter [11–13], described in the following.

The IMM Filter

The IMM filter produces a sub-optimal solution for a JMLS, which uses merging to reduce the number of branches in the model hypothesis tree. At each time instant, the filter approximates the posterior density by a Gaussian mixture, with only one Gaussian pdf connected to each model. As a basis for understanding the IMM filter, a block diagram describing one iteration of the algorithm is presented in Fig. 2.2. As seen in the figure, the filter is built up by a bank of M model-matched Kalman filters, where M is equal to three in the example.

At time $k-1$, the filter approximates the posterior density with a Gaussian mixture,

$$p\big(\mathbf{x}_{k-1}\big|\mathbf{Z}^{k-1}\big) \cong \sum_{i=1}^{M} w_{k-1}^{i}\mathcal{N}\big(\mathbf{x}_{k-1};\hat{\mathbf{x}}_{k-1|k-1}^{i},\mathbf{P}_{k-1|k-1}^{i}\big), \tag{2.31}$$

where the weight w_{k-1}^{i} describes the probability that model i is active,

$$w_{k-1}^{i} = \Pr\left\{m_{k-1}=i\big|\mathbf{Z}^{k-1}\right\}, \tag{2.32}$$

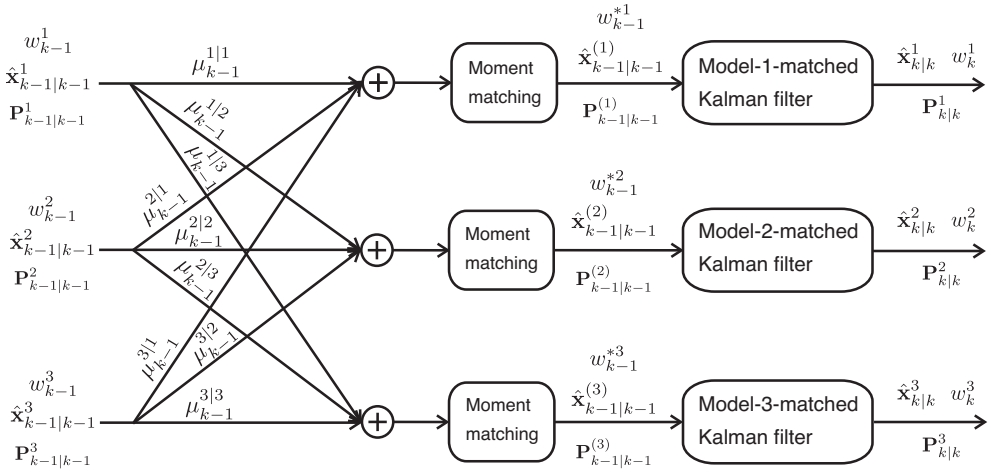

Figure 2.2: One iteration of the three-model IMM filter.

and the mean value $\hat{\mathbf{x}}^i_{k-1|k-1}$ is an estimate of the target state, given that model i was active in the last time interval. Further, the output from the filter is the weighted estimate

$$\hat{\mathbf{x}}_{k-1|k-1} = \sum_{i=1}^{M} w^i_{k-1} \hat{\mathbf{x}}^i_{k-1|k-1}, \tag{2.33}$$

and the weighted covariance matrix

$$\mathbf{P}_{k-1|k-1} = \sum_{i=1}^{M} w^i_{k-1} \Big[\mathbf{P}^i_{k-1|k-1} + \big(\hat{\mathbf{x}}_{k-1|k-1} - \hat{\mathbf{x}}^i_{k-1|k-1}\big)\big(\hat{\mathbf{x}}_{k-1|k-1} - \hat{\mathbf{x}}^i_{k-1|k-1}\big)^T \Big], \tag{2.34}$$

which are also the posterior mean and the posterior covariance matrix, respectively.

In the first step of the filter, the possible model switches of the system are considered. As seen in Fig. 2.2, for each model j, the input to its moment-matching block is a Gaussian mixture whose components depend on the mixture components at time $k-1$. The weights of the mixture, $\mu^{i|j}_{k-1}$, are called the *mixing probabilities*, and are defined as

$$\mu^{i|j}_{k-1} = \Pr\left\{ m_{k-1} = i \middle| m_k = j, \mathbf{Z}^{k-1} \right\}. \tag{2.35}$$

Using Bayes' formula, this is rewritten as

$$\mu^{i|j}_{k-1} = \frac{\pi_{ij} w^i_{k-1}}{\sum_{i=1}^{M} \pi_{ij} w^i_{k-1}} = \frac{\pi_{ij} w^i_{k-1}}{w^{*j}_{k-1}}. \tag{2.36}$$

In (2.36), π_{ij} is the transition probability between models i and j, w_{k-1}^{i} is the probability of model i before mixing, and w_{k-1}^{*j} is the probability of model j after mixing.

To limit the complexity of the filter, the input to the model-matched Kalman filters are single Gaussians that approximate the Gaussian mixtures from the switching step. The first two moments of these Gaussians, $\hat{\mathbf{x}}_{k-1|k-1}^{(j)}$ and $\mathbf{P}_{k-1|k-1}^{(j)}$, are found by moment matching,

$$\hat{\mathbf{x}}_{k-1|k-1}^{(j)} = \sum_{i=1}^{M} \mu_{k-1}^{i|j} \hat{\mathbf{x}}_{k-1|k-1}^{i} \tag{2.37}$$

$$\begin{aligned} \mathbf{P}_{k-1|k-1}^{(j)} = \sum_{i=1}^{M} \mu_{k-1}^{i|j} \Big[& \mathbf{P}_{k-1|k-1}^{i} + \\ & \big(\hat{\mathbf{x}}_{k-1|k-1}^{i} - \hat{\mathbf{x}}_{k-1|k-1}^{(j)}\big)\big(\hat{\mathbf{x}}_{k-1|k-1}^{i} - \hat{\mathbf{x}}_{k-1|k-1}^{(j)}\big)^{T} \Big]. \end{aligned} \tag{2.38}$$

After the model-matched filters have been applied, the output from the filter matched to model j is the posterior state vector estimate $\hat{\mathbf{x}}_{k|k}^{j}$, covariance matrix $\mathbf{P}_{k|k}^{j}$, and weight w_k^j, where the latter is calculated as

$$w_k^j = \frac{\Lambda_k^j w_{k-1}^{*j}}{\sum_{i=1}^{M} \Lambda_k^i w_{k-1}^{*i}}. \tag{2.39}$$

In (2.39), Λ_k^j is the model-conditioned likelihood function,

$$\Lambda_k^j = p\big(\mathbf{z}_k \big| m_k = j, \mathbf{Z}^{k-1}\big) = \mathcal{N}\big(\tilde{\mathbf{z}}_k^j; \mathbf{0}, \mathbf{S}_k^j\big), \tag{2.40}$$

where $\tilde{\mathbf{z}}_k^j$ and $\mathbf{S}_k^j$ are the innovation and the innovation covariance matrix, respectively, obtained by the model-matched Kalman filter.

2.2.2 Semi-Markov-Based Model Switching

In a discrete-time first-order Markov model, the transition probability π_{ij} depends only on the current state, and not on previous states. Further, the transition probability does not depend on for how long the current state has been active. The time spent in a model before a transition away from the model is called *sojourn time* [14], or holding time [15]. To have a transition probability that depends on this time, we extend the state vector to also include the sojourn time T_k

$$\overline{\mathbf{x}}_k = \begin{bmatrix} \mathbf{x}_k^T & T_k \end{bmatrix}^T. \tag{2.41}$$

The extended state vector $\overline{\mathbf{x}}_k$ now includes the target-related components such as, for example, position and velocity, the mode variable m_k, and the sojourn time T_k. By also having a model for the sojourn-time dependent transition probability,

$$\pi_{ij}\big(T_{k-1}\big) = \Pr\{m_k = j \big| m_{k-1} = i, T_{k-1}\}, \tag{2.42}$$

a switching system that depends on the holding time can be defined. Such a system is called a *semi-Markov* system.

There have been several algorithms that incorporate semi-Markov switching, and solve the conceptual problem in a sub-optimal manner. These are the sojourn-time dependent Markov (STDM) IMM algorithm [16, 17], the computationally less complex semi-Markov IMM (SM-IMM) algorithm [18], and the latest change moment testing (LCMT) algorithm [19].

Chapter 3

Data Association

In the previous chapter, we discussed the filtering part of a tracking algorithm, and concluded that additional problems need to be solved before filtering can take place. To update the predicted target states, we want to use information from the measurements received at the current instance of time. To do so, we need to find out which measurements that are generated by targets, and which are spurious. We then need to assign the former measurements to the corresponding targets. These two steps constitute the problem of data association. Ideally, each target gives rise to exactly one measurement, and there are no spurious measurements. However, for a number of reasons, the ideal scenario rarely occurs. First, all targets are not always detected, due to, e.g., power limitations and terrain obscuration. Second, false measurements arise due to receiver noise and signal reflections from uninteresting objects, or from the ground and surrounding terrain, called clutter. Third, the resolution capability of the sensor can be such that all targets are not always resolved, leading to fewer true detections than the number of targets; or, the size of a target is large compared to the resolution and hence may give rise to several detections. In this chapter, the data association problem for the first two problems are considered, while the background of the radar resolution issue is the topic of the next chapter.

To the tracking problem there are two accepted, and philosophically different, approaches: either, the interest lies in estimating where each individual target is, or in estimating where there are targets. Although seemingly identical, these two approaches are different and must be treated differently. The first, traditional, approach is to give each target an identity label, and to track the targets while trying to maintain the identities. In situations of targets being closely spaced, this is a delicate and intricate problem that may not be solvable. When target identities are not relevant to the user, the

approach to the tracking problem is different. One way to describe it is to use a random finite set description of the targets, which inherently has the orderless (label-free) property. In this chapter, these two approaches of target tracking are introduced, and the most common data association algorithms within each approach are described.

3.1 Tracking with Target Identity

In this section, we discuss the traditional approach to target tracking, *viz.* to assign each track an identity label, and to perform filtering and data association while trying to maintain the knowledge of which target is which. First, we present the conceptual solution to the problem, and then discuss practicable algorithms that approximate the optimal approach. The track labeling problem is also discussed in short.

3.1.1 Conceptual Data Association Solution

We start by shortly discussing one conceptual solution to the data association problem. If we let each possible assignment of measurements to targets represent a hypothesis, the set of hypotheses at a certain time can be ordered in a hypothesis tree. That is, at each time instant there might be several candidate measurements for each target. Each branch through the tree then represents a sequence of assignments, where only one sequence can be true. A branch also represents a sequence of state vectors, where the computation of the states relies on the specific data association being made. An example of a hypothesis tree for a single target is shown in Fig. 3.1. In the figure, the target was associated with detection 3 at time 1, with detections 0 or 1 at time 2, and so on. Here, an association with detection 0 means that the target was not detected.

To express the conceptual solution, we make three assumptions. First, we assume that a target generates at most one measurement. Second, we assume that the number of targets, M, is fixed and known. That is, we know that there are M targets present all the time. Through this simplifying assumption the problems of track initiation and deletion are avoided. Third, we assume that the target motion is described by a single motion model $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, and that the measurement process is captured by a single measurement model $p(\mathbf{z}_k|\mathbf{x}_k)$. The underlying assumption is that the targets move independently of each other and that their respective measurements are conditionally independent. All the target state vectors and all measurements

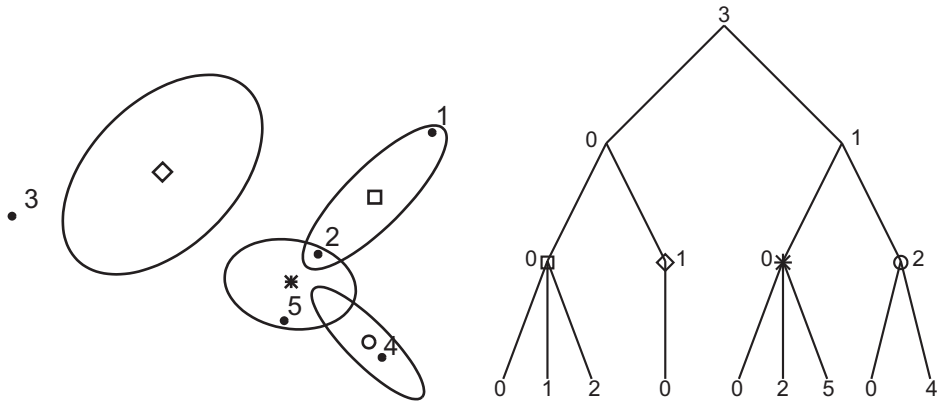

Figure 3.1: Example of a data association hypothesis tree for a single-target scenario. Left: predicted target states at time k obtained from the four track hypotheses at time $k-1$, the validation gate of each hypothesis (described later) and the received measurements at time k (marked with detection numbers). Right: the track hypothesis tree; the four possible hypotheses at time $k-1$ are marked with diamond, square, star and circle, and their respective state predictions are seen in the left figure.

are stacked

$$\mathbf{X}_k = \left[\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(M)}\right] \tag{3.1}$$

$$\mathbf{Z}_k = \left[\mathbf{z}_k^{(1)}, \ldots, \mathbf{z}_k^{(m_k)}\right], \tag{3.2}$$

where m_k is the number of measurements received at time k. Further, the history of received measurements are gathered in a set, $\mathbf{Z}^k$ (cf. (2.1)). The problem at hand is then to express the density $p(\mathbf{X}_k \big| \mathbf{Z}^k)$, i.e., the posterior density of the targets, given data up to the current time.

In order to express the probability density function of $\mathbf{X}_k$, we need to know which assignments that have been made during the course of time. We therefore introduce the assignment vector $\mathbf{r}_k^l = \begin{bmatrix} r_k^l(1) & r_k^l(2) & \cdots & r_k^l(M) \end{bmatrix}^T$ with the property that

$$r_k^l(i) = \begin{cases} j & \text{if detection } j \in \mathbf{Z}_k \text{ is associated with target } i \\ 0 & \text{if target } i \text{ did not generate a detection.} \end{cases} \tag{3.3}$$

Here $i = 1, \ldots, M$. We next construct a history of assignment vectors

$$\mathbf{R}_k^l = \left[\mathbf{r}_1^l, \mathbf{r}_2^l, \ldots, \mathbf{r}_k^l\right], \tag{3.4}$$

which represents the l^{th} path through a global hypothesis tree, where global indicates that the tree describes the associations of all targets. Conditioned on the assignment matrix $\mathbf{R}_k^l$, and assuming linear and Gaussian models, the posterior density $p(\mathbf{X}_k|\mathbf{Z}^k, \mathbf{R}_k^l)$ is Gaussian and can be obtained using a Kalman filter. The true posterior density $p(\mathbf{X}_k|\mathbf{Z}^k)$ is then the sum over all probability-weighted assignment-conditioned densities,

$$p(\mathbf{X}_k|\mathbf{Z}^k) = \sum_l \Pr\{\mathbf{R}_k^l|\mathbf{Z}^k\} p(\mathbf{X}_k|\mathbf{Z}^k, \mathbf{R}_k^l). \tag{3.5}$$

Since the number of assignment vectors grows exponentially with time, the conceptual solution is infeasible in practice. We are hence interested in simplifications that make data association practical.

Roughly speaking, there are three common ways of performing data association in practice:

1) Associating a single measurement to each target at each time step.

2) Assigning a weighted sum of measurements to each target at each time step.

3) Deferring the decision to a later time step, and aggregating all possible association hypotheses until that later time, where the best decision in retrospect is made.

The first two methods are referred to as single-hypothesis methods, while the latter describes the family of multiple-hypothesis methods. In the following, the main algorithms of the respective class of association schemes are described.

3.1.2 Nearest Neighbor Data Association

The simplest way to perform data association is to associate one measurement to each target at each time step, in the best possible manner. In this way, the single most likely measurement-to-target association hypothesis is propagated in time. In the hypothesis tree perspective, this is represented by a single branch running through the tree. The assignment made at a certain time might not at all be the best assignment in retrospect, which is one drawback with single-hypothesis methods.

The two most common single-hypothesis methods are the nearest neighbor (NN) [20] and the global nearest neighbor (GNN) [20] algorithms. The NN algorithm is the simplest of the two, and it finds the nearest neighboring

measurement of each target, under a specific distance measure. The measurement assigned to a target is then used for updating its predicted state. The nearest neighbor algorithm performs a local optimization in that it operates target by target. This can lead to several targets being associated with the same measurement, which might lead to poor tracking performance, due to track losses.

A global version of the NN algorithm is the GNN scheme. Instead of performing an optimization for each target, it searches for the best global association, considering all targets and measurements simultaneously. The global nearest neighbor problem can be formulated as a convex optimization problem and hence be solved using an efficient optimization algorithm, for example the Auction algorithm [21].

Although being superior to NN, the GNN algorithm still makes a hard decision on the associations, which under difficult scenarios with closely-spaced targets and a high number of false measurements can be a too crude approximation of the posterior density and hence perform poorly. Another property of data association algorithms that make hard decisions is that they suffer from the track repulsion effect [22] in dense target scenarios.

3.1.3 Probabilistic Data Association

Instead of associating the instantaneously single best measurement to each target, a second possibility is to use a weighted sum of the available measurements. Again, this can be done on a target-by-target basis, or by considering the global association problem. An algorithm for the local assignment problem is the probabilistic data association (PDA) algorithm [23], while the global problem is solved by the Joint Probabilistic Data Association (JPDA) algorithm [24]. These algorithms are multi-hypothesis per time step, in that several assignment hypotheses are considered per target, but single-hypothesis between time instances, due to the merging of all hypotheses per target into one. Hence, in the hypothesis tree perspective, this corresponds to a branching followed by an immediate merging of the branches into one. We here describe the JPDA algorithm in more detail, since it is an important component in both Paper II and Paper IV.

We start with the marginalization of the posterior density

$$p\big(\mathbf{X}_k\big|\mathbf{Z}^k\big) = \sum_{h\in\mathcal{H}} p\big(\mathbf{X}_k, h\big|\mathbf{Z}^k\big), \tag{3.6}$$

where h is a data association hypothesis in the set of all data association

hypotheses $\mathcal{H}$ at time k. The posterior density can be rewritten as

$$p\big(\mathbf{X}_k\big|\mathbf{Z}^k\big) = \sum_{h\in\mathcal{H}} \frac{p\big(\mathbf{Z}_k\big|\mathbf{X}_k, h, \mathbf{Z}^{k-1}\big)p\big(\mathbf{X}_k\big|\mathbf{Z}^{k-1}\big)}{p\big(\mathbf{Z}_k\big|\mathbf{Z}^{k-1}\big)} \Pr\big\{h\big|\mathbf{Z}^{k-1}\big\}. \tag{3.7}$$

By assuming linear and Gaussian models, and a Gaussian prior density $p\big(\mathbf{X}_k\big|\mathbf{Z}^{k-1}\big)$ (given by the JPDA approximation at time index $k-1$), the above density is a Gaussian mixture,

$$p\big(\mathbf{X}_k\big|\mathbf{Z}^k\big) = \sum_{h=1}^{N_\mathcal{H}} \beta_h \mathcal{N}\big(\mathbf{x}_k; \hat{\mathbf{x}}^h_{k|k}, \mathbf{P}^h_{k|k}\big), \tag{3.8}$$

where $N_\mathcal{H}$ is the total number of hypotheses, and β_h is the mixture weight, defined below.

The JPDA filter performs the following steps:

1. Formulate all global data association hypotheses, $\mathcal{H}$, which describe possible origins of $\mathbf{Z}_k$.

2. For each data association hypothesis $h \in \mathcal{H}$, update the predicted density for each target i with the assigned measurement j using a Kalman filter. The output of the filter is the mean value $\hat{\mathbf{x}}^{i,h}_{k|k}$ and covariance matrix $\mathbf{P}^{i,h}_{k|k}$.

3. Calculate the weight β_h of each mixture component h.

4. Use moment matching to approximate the Gaussian mixture by a single Gaussian.

The mixture weights β_h are given by

$$\beta_h = \frac{\bar{\beta}_h}{\sum_{h=1}^{N_\mathcal{H}} \bar{\beta}_h}, \tag{3.9}$$

$$\bar{\beta}_h = P_c(M_c)\frac{(M_k - M_t)!}{M_K!}\frac{1}{\big|\mathrm{FoV}\big|^{M_c}} \prod_{\mathcal{S}^h_u} \big(1 - P_D\big) \cdot \prod_{\{i,j\}\in\mathcal{S}^h_a} P_D\, g_{ij}, \tag{3.10}$$

where $P_c(M_c)$ is the probability of receiving M_c clutter detections, M_k is the total number of detections, and M_t is the number of target-generated detections. Further, a constant detection probability, P_D, is assumed, $\mathcal{S}^h_u$ is the set of unassigned targets, and $\mathcal{S}^h_a$ is a set including the pairs of detected

targets, i, and their assigned measurements, $\mathbf{z}_k^j$. Additionally,

$$\boldsymbol{\nu}_{ij} = \mathbf{z}_k^j - \mathbf{H}_k \mathbf{x}_{k|k-1}^{(i)}, \quad d_{ij}^2 = \boldsymbol{\nu}_{ij}^T \mathbf{S}_k^{-1} \boldsymbol{\nu}_{ij} \tag{3.11}$$

$$g_{ij} = \mathcal{N}\big(\boldsymbol{\nu}_{ij}; \mathbf{0}, \mathbf{S}_k\big) = \frac{1}{\big|2\pi \mathbf{S}_k\big|^{1/2}} e^{-\frac{d_{ij}^2}{2}}. \tag{3.12}$$

As described above, the first three steps of the JPDA algorithm describe the calculation of the components in the Gaussian mixture (3.8), while the final step is to approximate that Gaussian mixture density for each target with a single Gaussian. This is done by second-order moment matching, i.e.,

$$p\big(\mathbf{x}_k^{(i)} \big| \mathbf{Z}^k\big) \cong \mathcal{N}\big(\mathbf{x}_k^{(i)}; \hat{\mathbf{x}}_{k|k}^i, \mathbf{P}_{k|k}^i\big), \tag{3.13}$$

where[1]

$$\hat{\mathbf{x}}_{k|k}^i = \sum_{h=1}^{N_{\mathcal{H}}} \beta_h \hat{\mathbf{x}}_{k|k}^{i,h} \tag{3.14}$$

$$\mathbf{P}_{k|k}^i = \sum_{h=1}^{N_{\mathcal{H}}} \beta_h \left\{ \mathbf{P}_{k|k}^{i,h} + \big(\hat{\mathbf{x}}_{k|k}^i - \hat{\mathbf{x}}_{k|k}^{i,h}\big)\big(\hat{\mathbf{x}}_{k|k}^i - \hat{\mathbf{x}}_{k|k}^{i,h}\big)^T \right\} \tag{3.15}$$

are the mean and covariance matrix of the single Gaussian.

The JPDA algorithm is often described in an alternative, but equivalent, fashion [20], where the computation of the state estimates includes the calculation of a weighted measurement residual, which is used in an ordinary Kalman filter update.

Both the PDA and the JPDA filter have a tendency to merge tracks when the targets are closely spaced. This is referred to as the track coalescence effect [25]. In Paper II, an adjusted version of the JPDA filter is presented, which does not experience track coalescence. Another adjustment of the JPDA filter to avoid track coalescence is the JPDA* algorithm [26].

3.1.4 Multi-Hypothesis Tracking

The two previously discussed approaches to data association are either to assign the (globally) best measurement to each target, or to assign a weighted sum of measurements to each target, where a weight depends on the probability of the corresponding data association hypothesis. A third alternative is to wait with the assignment, and instead keep several data association

[1] Although the sums in (3.14) and (3.15) can be done as written, in practice there would be a step of marginalization over the single-target association events.

hypotheses, and aggregate them over time. Then, a hypothesis describes a possible *sequence of assignments* for each target. This is the foundation of Multi-Hypothesis Tracking (MHT) [20]. MHT is not a single algorithm, but a family of algorithms that maintain multiple data association hypotheses.

Multi-hypothesis tracking is a deferred decision logic in the sense that the data association at a time k is not set until at a later time step $k+N$ ($N > 0$), at which time the availability of more data increases the probability that the true hypothesis at time k is retained. In each time step, MHT forms a set of plausible association hypotheses. Over time, the hypotheses will build up a hypothesis tree for each target, in the same way as for the conceptual solution to data association in Section 3.1.1. This makes MHT different to the single-hypothesis algorithms, as it propagates more than one hypothesis in time. Since the hypothesis tree grows with time, a reduction of branches is performed by discarding the least likely, or least probable, hypotheses at each time step. The original formulation of MHT was given in [27], and since then many different formulations of MHT has been given, e.g., in [20, 28–35]. Compared to a conventional single-hypothesis tracking algorithm, MHT can handle 10-100 times higher false-alarm densities [20]. MHT is also better suited than conventional algorithms when the target density is high. If the MHT algorithm is probability-based, another advantage with MHT is that the algorithm can tell the user how certain it is about the existence of currently presented tracks.

There are two fundamentally different approaches to MHT. The first one is *hypothesis-oriented* MHT, which is the one described in the original MHT formulation [27]. The second one is *track-oriented*, and it is described, e.g., in [20, 28–31]. We will give a short introduction to both approaches, starting with track-oriented MHT.

Track-Oriented MHT

In the track-oriented approach, as the name implies, we work on a per-track perspective. In our description, a track constitutes the description of a possible target, and each track is represented by a hypothesis tree. A branch running through the tree from top to bottom is referred to as a track hypothesis. In other descriptions, e.g., in [20] a track is a sequence of detections associated to a target, i.e., what we call a track hypothesis.

At each time step, all n possible associations of measurements to a certain track hypothesis are used to create n new branches of that track hypothesis. Consider the example in Fig. 3.1, where we have a single-target scenario, for which the target is described by a single validated track. To the left in the figure, we have plotted the track hypotheses predictions at time k, for each of

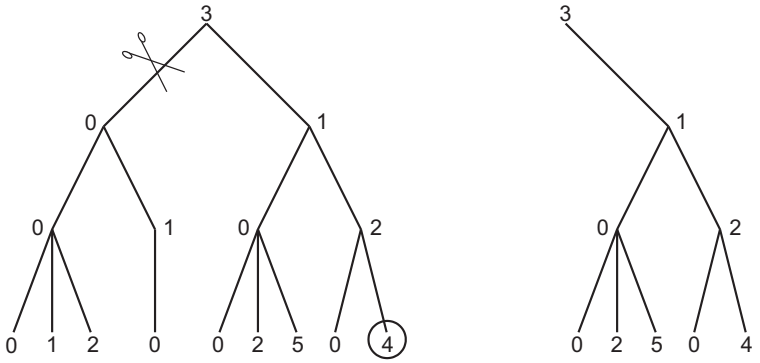

Figure 3.2: Example of N-scan pruning ($N = 2$) for a single-target scenario. To the left we have the hypothesis tree before N-scan pruning, and to the right we have the corresponding tree after pruning. The best track hypothesis (in terms of track score or probability) at time k is marked with a circle. All track hypotheses with the same root node $N = 2$ steps back (detection 1) are to be kept, whereas the rest are pruned. In the example, the hypotheses which have detection 0 as root node $N = 2$ steps back are thus removed.

the four track hypotheses at time $k-1$, and the gate of each such hypothesis. In the figure, we have also plotted the five received measurements at time k. To the right, the hypothesis tree is depicted, where the marked nodes of the tree represent the four hypotheses at $k-1$. Each hypothesis prediction is associated with all measurements within its gate. For each association, a new branch of the tree is created. The numbers next to the branches declare the measurement number. Recall that the number 0 branches represent the hypothesis that the target is not detected at time k.

Since the number of hypotheses grows exponentially over time, the number of hypotheses must be reduced to obtain a practicable approach. This can be done by merging similar branches together, or by pruning low-probability branches, or by performing both merging and pruning. One way of performing pruning is N-scan pruning [36], which is illustrated in Fig. 3.2. The pruning is either based on track hypothesis probability or likelihood.

In the multi-target case, the track-oriented MHT is more complicated, since we also need to consider possible conflicts between targets, where a conflict occurs if two tracks are associated with the same detection. To treat conflicts, the concept of a global hypothesis is introduced. Consider the two-target example of Fig. 3.3. A few of the conflicts between the hypothesis trees are marked with circles, and the globally possible data association

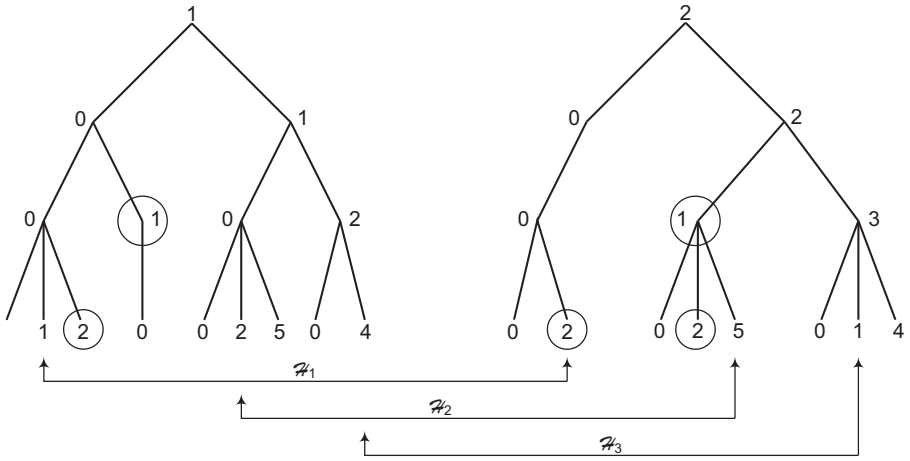

Figure 3.3: Example of hypothesis trees for a two-target scenario. Each track represents one target. Three of the global hypotheses are marked as $\mathcal{H}_1$ to $\mathcal{H}_3$, and some conflicts are marked with circles.

hypotheses are the ones where there are no conflicts between the targets. Those hypotheses are referred to as feasible global hypotheses. In Fig. 3.3, three of the feasible global hypotheses are marked with arrows. As seen, a global hypothesis contains one track hypothesis from each track. Note that missed-detection hypotheses are never in conflict. If there are L tracks, a global hypothesis $\mathcal{H}_k$ is described as

$$\mathcal{H}_k = \{h_{1i_{1k}}, h_{2i_{2k}}, \ldots, h_{Li_{Lk}}\}, \tag{3.16}$$

where $h_{li_{jk}}$ represents the track hypothesis i_j of hypothesis tree l, under the global hypothesis k.

Just as for the single-target case, the hypothesis trees need to be reduced for the algorithm to be feasible. The N-scan pruning approach is applicable also to the multi-target case, where the global hypotheses are used instead of the local ones. The pruning algorithm then finds the best global hypothesis, $\mathcal{H}^*$, and prunes the hypothesis trees according to that hypothesis.

Hypothesis-Oriented MHT

The original MHT formulation [27] was hypothesis-oriented. This means that it works directly on the feasible global hypotheses. So at time k, the set of global hypotheses from time $k-1$ are expanded to consider the measurements received at time k. Each global hypothesis from the previous time instant is hence expanded into a set of global hypotheses, which are feasible with

respect to conflicts between the targets that are part of the hypothesis. Just as for track-oriented MHT, pruning is used to reduce the number of global hypotheses.

According to [20], a problem with the original MHT formulation is that a lot of low-probability hypotheses are created, and then immediately discarded due to their low probability. This wastes computational resources. However, the problem is relieved in [32] through the use of an extended version of Murty's method [37].

3.1.5 Track Handling

In the following description, a track constitutes a description of the states of a target over time. Ideally, there is a one-to-one correspondence between a target and a track, but due to, e.g., clutter and missed detections, it is not always the case in practice. The track handling of a tracking filter has the role of initiating, validating and deleting tracks. A simple way of track validation is the M/N validation rule, which states that if a track has been assigned to a measurement in M of the last N time steps, it is considered to represent a target. A way of determining the quality of a track is the concept of track score [20]. The track score is essentially a hypothesis test, with hypotheses:

$\mathcal{H}_0$: The track is due to clutter.

$\mathcal{H}_1$: The track represents a target.

The track score (TS) is the quotient of the probability of the true track hypothesis, given data $\mathbf{Z}^k$, to the false-track hypothesis probability

$$TS(k) = \frac{\Pr\left\{\mathcal{H}_1|\mathbf{Z}^k\right\}}{\Pr\left\{\mathcal{H}_0|\mathbf{Z}^k\right\}} = \frac{p\big(\mathbf{Z}^k|\mathcal{H}_1\big)\Pr\{\mathcal{H}_1\}}{p\big(\mathbf{Z}^k|\mathcal{H}_0\big)\Pr\{\mathcal{H}_0\}}, \tag{3.17}$$

where $\Pr\{\mathcal{H}_i\}$ is the a-priori probability of hypothesis $\mathcal{H}_i$, $i=0,1$. Assuming independent measurements from scan to scan, the track score can be written on a recursive form

$$TS(k) = \frac{p\big(\mathbf{Z}_k|\mathcal{H}_1\big)}{p\big(\mathbf{Z}_k|\mathcal{H}_0\big)} TS(k-1). \tag{3.18}$$

Often the logarithm of the track score is used, which makes the score update a simple summation. The track score can be used as input to a Sequential Probability Ratio Test (SPRT) [38], which compares the track score to an upper and a lower threshold. If the score is above the upper threshold,

the track is validated, whereas if the score of a track falls below the lower threshold it is declared false and the track is deleted. When the score is in between the thresholds, it is tentative.

Gating

If the clutter level is high, and the number of targets is large, there will be many possible data associations for each track. Several of these associations may be very unlikely, especially if the surveillance area is large. Since those unlikely events will not affect the description of the posterior density, they can be removed. One method for avoiding the calculation of unlikely hypotheses is gating. With gating, an acceptability region is formed around each track estimate, where measurements that fall within the region are considered, whereas measurements outside the region are ignored. The gate size is often selected to include the true target-generated detection with some probability. Except for being a method to reduce complexity, gating also makes sure that an unlikely detection is not associated to a target, when the target is not detected. The gating procedure is illustrated in Fig. 3.1. For more information, see [20].

3.2 Tracking Without Target Identity

In this section, we consider the problem of target tracking without target identity. That is, we seek to estimate where there are targets, but are not interested in which one is which. Examples of applications where such problems arise are collision avoidance in automotive systems, radar cueing, and threat neutralization. The conceptual solution to this problem can be approached by means of Finite Set Statistics (FISST) [39–41] by Mahler, which provide a way to extend the single-target single-sensor Bayes statistics to the multi-target multi-sensor realm. Apart from describing the conceptual solution, we provide an overview of multiple target tracking and data association without target identity, using FISST-based methods. Note, however, that in the practical algorithms that are based on FISST, target labeling can be performed in retrospect using an over-head labeling algorithm [42, 43]. In Paper II, the relation between FISST and ordinary statistics is utilized to improve traditional target tracking filters, such as JPDA, to the problem of tracking targets without identity.

Before turning to the practical algorithms, we first describe the notion of random finite sets, the problem formulation, and its conceptual solution.

3.2.1 Random Finite Sets

A random finite set (RFS) is a generalization of a random matrix. To connect to tracking, say that we have three targets with state vectors $\mathbf{x}_k^{(1)}$, $\mathbf{x}_k^{(2)}$ and $\mathbf{x}_k^{(3)}$, which are all unknown and stochastic. That is, we know the number of targets, but not their states. If the state vectors are of the same dimension, they can be collected in a matrix $\mathbf{X}_k$, as described in (3.1). The matrix is random, and it has a probability density function which is given by $p\big(\mathbf{X}_k\big)$. Assume now that we do not know how many targets there are in the scene, but we still want to be able to say something about the number of targets and their respective states. One way of doing this is to describe the targets by a random finite set (RFS). The formal definition of an RFS Ξ_k is [40]

Definition An RFS Ξ_k is a random variable that draws its instantiations $\Xi_k = X_k$ from the hyperspace $\mathcal{F}(\mathcal{X})$ of all finite subsets X_k of some underlying space $\mathcal{X}$.

In target tracking it is most common that $\mathcal{X} = \mathbb{R}^n$, where n is the dimension of the state vector $\mathbf{x}_k$. Possible outcomes of the RFS are then

$$X_k = \begin{cases} \emptyset & \text{if no target is present} \\ \{\mathbf{x}_k\} & \text{if there is one target with state vector } \mathbf{x}_k \text{ present} \\ \left\{\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)}\right\} & \text{if there are two targets with states } \mathbf{x}_k^{(1)} \text{ and } \mathbf{x}_k^{(2)}, \end{cases}$$

and so on. In the following, we will refer to the outcome X_k as the RFS. Note that the set is *without ordering*, which means that, for two targets,

$$X_k = \left\{\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)}\right\} = \left\{\mathbf{x}_k^{(2)}, \mathbf{x}_k^{(1)}\right\}.$$

Just as for random vectors and matrices, there exists a probability density function $p\big(X_k\big)$ of an RFS, X_k. Due to the inherent orderless property of RFSs, its relation to the joint pdf of ordered target vectors is

$$p\Big(\left\{\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n)}\right\} \Big| n\Big) = n! p\big(\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n)}\big), \tag{3.19}$$

since the n elements of the RFS can be permuted in $n!$ ways. The number of elements of a set is called *cardinality*, and is written $\big|X_k\big|$. For an RFS, the cardinality is a discrete random variable. An RFS is hence described by its cardinality distribution, and the distribution of its elements.

3.2.2 Problem Formulation for Tracking Without Target Identity

The general multi-target multi-sensor tracking problem in the FISST framework is to estimate the target RFS, given all data up to the current time, where the RFSs are defined as follows.

X_k is the target RFS, namely the set of all targets present at time k (if any),

Z^k is the measurement history RFS, i.e., the set of all measurements from time 1 to time k.

We thus seek to calculate $p\big(X_k|Z^k\big)$, analogously to the single-target case.

3.2.3 Conceptual Solution: The Multi-Target Bayes Filter

For a single target, the solution to the tracking problem is given by the single-target Bayes filter, see (2.5). For multiple targets, FISST enables the derivation of an analogous multi-target expression. By rewriting the multi-target posterior pdf using Bayes' rule,

$$p\big(X_k|Z^k\big) = \frac{p\big(Z_k|X_k, Z_{1:k-1}\big)p\big(X_k|Z_{1:k-1}\big)}{p\big(Z_k|Z_{1:k-1}\big)}, \tag{3.20}$$

we obtain a recursive method for evaluating the density, given the multi-target likelihood $p\big(Z_k|X_k, Z_{1:k-1}\big)$ (multi-target measurement model) and the multi-target prior $p\big(X_k|Z_{1:k-1}\big)$. Similar to the Chapman-Kolmogorov equation (cf. (2.8)) for the single-target Bayes filter, we have [40]

$$\begin{aligned} p\big(X_k|Z_{1:k-1}\big) &= \int p\big(X_k|X_{k-1}, Z_{1:k-1}\big)p\big(X_{k-1}|Z_{1:k-1}\big)\delta X_{k-1} \\ &= \int p\big(X_k|X_{k-1}\big)p\big(X_{k-1}|Z_{1:k-1}\big)\delta X_{k-1} \end{aligned} \tag{3.21}$$

where the integral is a *set integral* [40]. As described in [40], the multi-target Bayes filter is mathematically and practically intractable in general, since analytical expressions are difficult to find, and set integrals are computationally very heavy. Since none of the work in this thesis uses set integration, the concept is not further discussed. In (3.21), the multi-target density $p\big(X_k|X_{k-1}\big)$ is the multi-target analog of the single-target motion model [40], which is assumed to fulfill the Markov property of a random process [2].

The single-target Bayes filter is computationally heavy, except for some special cases such as linear Gaussian models, which leads to the Kalman filter. The multi-target Bayes filter is even more computationally cumbersome, due to the set integrals, whereby reasonable approximations have to be made in order to obtain practical filters.

3.2.4 Probability Hypothesis Density

The Kalman filter recursively estimates the posterior expected value and covariance matrix of a state vector $\mathbf{x}_k|\mathbf{Z}^k$. To find a corresponding multi-target filter for tracking without target identity is not straightforward. The first problem one faces is the question of what the expected value of a random finite set X_k is. The purpose of this section is to shortly answer that question.

Since addition of sets is undefined, the expected value of an RFS cannot be defined equivalently to its vector counterpart. Instead, an indirect expected value is defined through a transformation $X_k \mapsto T_{X_k}$, which maps the RFS X_k into vectors T_{X_k} in a vector space [40]. The mapping should preserve the set-theoretic structure in that unions be transformed into sums. According to [40], it is common practice in the point process literature [44] to choose

$$T_{X_k} = \delta_{X_k}(\mathbf{x}_k), \tag{3.22}$$

where

$$\delta_{X_k}(\mathbf{x}_k) \triangleq \begin{cases} 0 & \text{if } X_k = \emptyset \\ \sum_{\mathbf{w}\in X_k} \delta_{\mathbf{w}}(\mathbf{x}_k) & \text{otherwise.} \end{cases} \tag{3.23}$$

For each element $\mathbf{x}_k$ in X_k, the transformation thus puts a Dirac point mass in that point $\mathbf{x}_k$ of the single-target state space. The indirect multi-target expected value $v_{k|k}(\mathbf{x}_k)$ of the RFS $X_k|Z^k$, given by the expected value of $\delta_{X_k}(\mathbf{x}_k)$, is thus

$$v_{k|k}(\mathbf{x}_k) \triangleq \mathbb{E}\left[\delta_{X_k}(\mathbf{x}_k)\right] = \int \delta_{X_k}(\mathbf{x}_k) p\big(X_k|Z^k\big) dX_k. \tag{3.24}$$

As we see, the first-order statistical moment of the RFS X_k is a function on the single-target state space. Actually, it is an intensity function, with the property that $v_{k|k}(\mathbf{x}_k)d\mathbf{x}_k$ gives the expected number of targets in a small area $d\mathbf{x}_k$. The function is called the *probability hypothesis density* (PHD). It is also referred to as the target intensity function. The concept of a PHD was, according to [45], first discussed in some unpublished work [46], [47]. The proof that the PHD is indeed a first-order statistical moment was first given in [45].

An intuitive understanding of the PHD can be given by the following property of the intensity function. Consider an area $\mathcal{S}$ in the single-target state-space. The expected value of the number of targets in the RFS X_k, which reside within the area, is

$$\mathbb{E}\left[|X_k \cap \mathcal{S}|\right] = \int_{\mathcal{S}} v_{k|k}(\mathbf{x}_k) d\mathbf{x}_k. \tag{3.25}$$

All tracking algorithms based on FISST assume that the random finite sets are of a certain type. In the following subsections we will introduce these types, and their implications.

Poisson RFS

A Poisson distribution is completely determined by its expected value (the variance being equal to the expected value). In the same sense, a Poisson RFS is completely characterized by its intensity function $v(\mathbf{x})$. An RFS X is a Poisson RFS if its cardinality distribution $\Pr\{\left|X\right| = n\}$ is a Poisson distribution with expected value $\hat{N} = \int v(\mathbf{x})d\mathbf{x}$, and if its elements are independent and identically distributed [40], according to the probability density $p\big(\mathbf{x}\big) = v\big(\mathbf{x}\big)/\hat{N}$.

Cluster RFS

A cluster RFS is a generalization of the Poisson RFS [40]. An RFS X is a cluster RFS if its elements, for finite cardinality, are independent and identically distributed according to $v\big(\mathbf{x}\big)/\hat{N}$. In difference to the Poisson RFS, the cardinality distribution is arbitrary. Assuming independent targets with identical distribution is not always a good approach, which is a weakness of some of the FISST-based filters.

3.2.5 The Probability Hypothesis Density Filter

The PHD, as seen in (3.24), is the first-order moment of a random finite set, corresponding to the expected value of a stochastic vector. Indeed, if an RFS X_k contains only one target vector $\mathbf{x}_k$, its PHD $v_{k|k}(\mathbf{x}_k)$ is the probability density function $p(\mathbf{x}_k|\mathbf{Z}^k)$ of that vector. Under the assumption that the main part of the information regarding the RFS is given by its first-order moment, a recursive filter for estimating the PHD would be of practical value. This is the foundation for the PHD filter [48], which is discussed here. Relating to the single-target problem, the PHD filter is the multi-target analog of

fixed-gain filters, such as the (α-β-γ) filter [49], which propagate the first-order moment of a state vector $\mathbf{x}_k$ in time. Just as the Kalman filter, the PHD filter has a prediction and a measurement update step. We will not derive the filter here, nor state the filter equations, but will only discuss its assumptions, and some of their implications. For filter equations of the PHD filter, see for example [48, 50].

The PHD filter was originally intended for cluster tracking and group-target tracking applications, but has shown to be of practical interest beyond these areas [51].

Assumptions

The PHD filter resides on the following assumptions:

1. Each target evolves and generates measurements independently of the other targets.

2. Clutter is Poisson distributed and independent of target-generated measurements.

3. The predicted multi-target RFS is a Poisson RFS.

The first two assumptions are standard for tracking applications, whereas the last one is PHD-specific. That the predicted RFS is a Poisson RFS implies that the cardinality of the set at prediction is Poisson distributed and that the targets are independent and identically distributed at prediction. According to [48], it is a reasonable assumption if interactions between targets are negligible. However, it leads to the problem that the cardinality estimate drops quickly if some targets are not detected for a number of time steps, which is called the missed-detection problem [52].

To model increasing and decreasing number of targets, the PHD filter has two steps called target birth and target death. In the PHD filter setting, target birth appears in two different forms: either by spontaneous birth or by spawning from a target at the previous time instant. Further, disappearance of targets from the visibility region of the sensor is modeled by a death process. With a certain probability, given by the death process, a target vanishes, or dies, at the next time step.

Although expressions for the PHD filter exist, they are not closed form in general. Numerical integrations are possible, but usually intractable. Two practicable implementations of the filter are particle filtering [53], and a Gaussian-mixture approximation [54].

Appealing properties of the PHD filter are that it avoids all explicit multi-target to multi-detection assignments, that its complexity is linear in both

number of targets and number of measurements, and that it is easy to implement. The main drawback with the PHD filter is the assumption that the target RFS is Poisson, which is unintuitive and leads to peculiar behavior of the filter; see for example [52].

3.2.6 The Cardinalized Probability Hypothesis Density Filter

A drawback with the PHD filter is that it represents a target RFS with only its first-order moment. In principle, one would instead want a Kalman-filter equivalent for multi-target problems, which requires a description of the second-order moment of an RFS. Theoretically, according to [51], it is indeed possible to find such a filter, but it is claimed not to be practically implementable. Instead, to improve upon the PHD filter in practice, a better description of the cardinality distribution is a feasible approach. In [51, 55] such an approach is devised, where not only the first-order statistical moment of the multi-target RFS is propagated, but also its full cardinality distribution $\Pr\{n_k|Z^k\}$, where n_k is the cardinality of the set. The method is called the *Cardinalized* Probability Hypothesis Density (CPHD) filter. With the improved cardinality representation, the CPHD filter provides lower-variance estimates of the number of targets.

The CPHD filter rests on the following assumptions:

1. Each target evolves and generates measurements independently of the other targets.
2. Clutter is a cluster RFS, and independent of the object generated RFSs.
3. Predicted and posterior target RFSs are approximated as cluster RFSs.
4. The birth RFSs are cluster RFSs, and independent of the surviving target RFSs.

If we compare the above assumptions to those of the PHD filter in Section 3.2.5, we see that the Poisson RFS assumptions on target, clutter and birth RFSs have been exchanged for cluster RFSs. Through this, the Poisson distribution of the number of elements in a set is exchangeable for an arbitrary probability mass function. For the target RFS we are then allowed to propagate the full cardinality distribution $\Pr\{n_k|Z^k\}$ in time, to provide better (lower-variance) estimates of target cardinality. However, the assumption that the targets are identically distributed is still present. Often, for clutter and birth processes, the Poisson RFS assumption is used also in the CPHD setting.

The original CPHD derivation uses FISST [51], but the filter has also been derived in two different ways using ordinary statistics, in [56] and [57]. Different formats of the prediction and measurement update equations are found in [51, 56, 58].

Two practical implementations of the CPHD filter have been proposed. The first one is a sequential Monte Carlo approach [40]—a particle filter—and the second a Gaussian-mixture filter, called GM-CPHD [58, 59], which provides a closed-form solution to the problem under linear-Gaussian conditions. Ground target tracking using GM-CPHD, together with road information, is described in [60].

The output of the CPHD filter is an estimate of the cardinality of the target RFS, as well as the state vectors of those targets without ordering. There are several possible cardinality estimates to choose from, where the most commonly used one is the maximum a posteriori (MAP) estimate

$$N_{k|k}^{\mathrm{MAP}} = \arg\max_{n_k} \Pr\{n_k | Z^k\}. \tag{3.26}$$

The MAP approach gives stable estimates, which are integer, and it can be directly used for extracting the state vectors.

In the state vector output, we want to find the state vectors that correspond to the expected number of targets. For the continuous CPHD, target concentrations are seen as peaks in the intensity function. In the Gaussian mixture case, these peaks are located in the centers of the largest mixture components. The state vector estimates are hence the state vectors corresponding to the $N_{k|k}^{\mathrm{MAP}}$ mixture components with largest weights. In particle filter implementations, state vector extraction is more complicated, and has to rely on an efficient clustering of particles into an appropriate number of clusters, e.g., the cardinality MAP estimate [58]. In dense target situations, where no natural clustering can be done, the state estimates will be unreliable. Thus, the avoidance of clustering is a major benefit of the Gaussian mixture CPHD.

In Fig. 3.4, we give an example of how the intensity function $v_{k|k}(\mathbf{x}_k)$ may look in practice. The scenario is a ground target tracking scenario with nine vehicles moving closely together. Simulated measurements are provided by an airborne radar measuring range, azimuth and elevation. The measurement model is non-linear, so tracking is performed using the GMCPHD filter together with an Extended Kalman Filter (EKF). At a certain time, the filter provides an estimate of the intensity function expressed by the posterior Gaussian mixture. The posterior intensity function $v_{k|k}(\mathbf{x}_k)$ is here described by 52 Gaussian components. In the example, the MAP estimate of the cardinality is 8 targets.

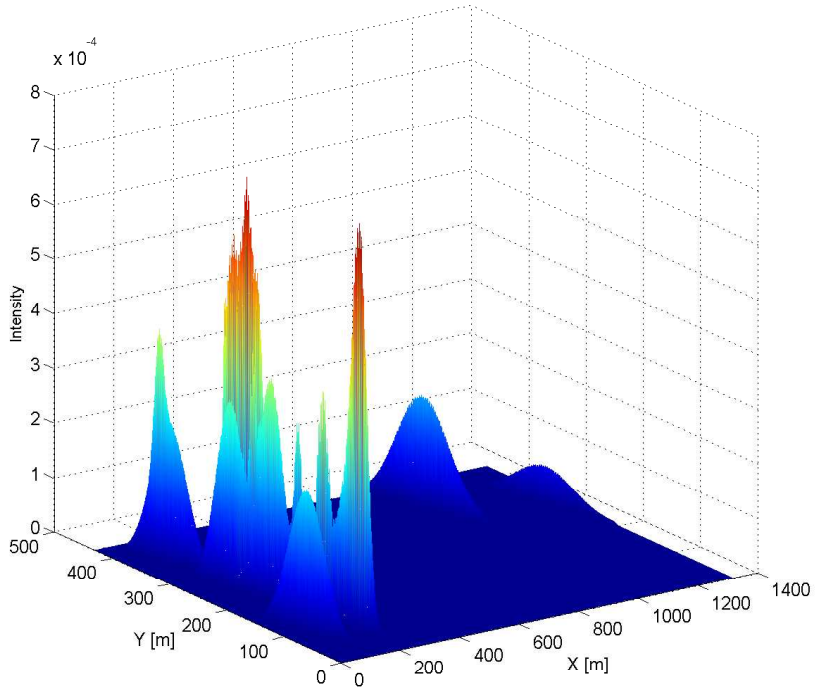

Figure 3.4: Example of a PHD surface (intensity function) for a ground target tracking scenario, with an airborne radar approaching the targets. The number of targets in the scene is 9 and the CPHD MAP estimate at the current time is 8. The intensity function is a sum of 52 mixture components. The sensor has good range accuracy, but poorer angular accuracy. Thus, the mixture components are narrow in the range direction. The PHD surface is plotted in the (x, y) plane. The scale on the x- and y axes are in meters, where the coordinate system is centered around the mid point of the scenario.

The complexity of the CPHD filter is linear in number of targets, but of higher complexity in number of measurements, depending on how the filter is implemented. For a straightforward implementation, the filter is cubical in the number of measurements. For more information, see [58]. Just as for all FISST-based filters, the labeling of targets cannot be performed in the standard CPHD filter. However, for the GMPHD filter, the labeling scheme proposed in [61] is applicable. Each mixture component then has its

own label. When mixture components are merged, the component with the largest weigh determines the label of the merged component.

The CPHD filter solves the missed-detection problem [52] of the PHD filter. However, in [62] a different missed-detection problem for CPHD was observed, called spooky action at a distance.

3.3 Performance Measures of Multi-Target Filters

To be able to assess the performance of a filter, a measure of performance is essential. For single-target scenarios, or scenarios with known number of targets, the root-mean square error (RMSE) is a standard measure of performance. If the true state at time k is $\mathbf{x}_k^{\text{true}}$, and if we from a Monte Carlo simulation of a filter obtain N estimates $\hat{\mathbf{x}}_k^l, l = 1, \ldots, N$ of the state at time k, the accuracy of the estimates is given by the RMSE

$$e^{\text{RMSE}}(k) = \sqrt{\frac{1}{N}\sum_{l=1}^{N}\left(\hat{\mathbf{x}}_k^l - \mathbf{x}_k^{\text{true}}\right)^T\left(\hat{\mathbf{x}}_k^l - \mathbf{x}_k^{\text{true}}\right)}. \tag{3.27}$$

For multi-target tracking algorithms, however, there is no standard performance measure. The problem can be illustrated with an example. Say that we have a one-dimensional problem, and that we apply two different tracking filters on the problem. The true position of the only target is $x^{\text{true}} = 10$. Say further that Filter 1 tells us that there are two targets, one with state $\hat{x}_{f_1}^{(1)} = 1$ and the other one with state $\hat{x}_{f_1}^{(2)} = 9.9$, and that Filter 2 tells us that there is only one target, with state $\hat{x}_{f_2}^{(1)} = 4$. Which of the filters then produce the best estimate at the current time? One of the estimates of filter 1 is very close to the truth, but the cardinality is wrong. Filter 2 has the correct cardinality, but a poor estimate of the position. Issues of this kind makes multi-target performance measures tricky. Often, a set of measures are required to capture strengths and weaknesses of different algorithms.

A first attempt to construct a single performance measure for multi-target scenarios was given in [63], with a Wasserstein-based [64] measure of performance. Criticism towards the measure is found in [65, 66], where a new performance measure, called Optimal Subpattern Assignment (OSPA), is introduced. The OSPA measure is also based on the Wasserstein distance, and the measure is an update of the measure in [63]. The OSPA measure tries to capture the quality of both the cardinality and the state estimation. Let $\mathbf{X}$ be the set of true target states and $\widehat{\mathbf{X}}$ be the set of target estimates, with

cardinalities n and m, respectively. The OSPA measure $\bar{d}_p^{(c)}$ is then defined as

$$\bar{d}_p^{(c)}(\widehat{\mathbf{X}}, \mathbf{X}) = \left(\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^{m} d^{(c)}(\mathbf{x}_i, \hat{\mathbf{x}}_{\pi(i)})^p + c^p(n-m) \right) \right)^{1/p}, \qquad (3.28)$$

if $m \leq n$ and $\bar{d}_p^{(c)}(\widehat{\mathbf{X}}, \mathbf{X}) = \bar{d}_p^{(c)}(\mathbf{X}, \widehat{\mathbf{X}})$ otherwise. In (3.28), $d^{(c)}(\mathbf{x}, \hat{\mathbf{x}}) \triangleq \min(c, d(\mathbf{x}, \hat{\mathbf{x}}))$ is the distance d between $\mathbf{x}$ and $\hat{\mathbf{x}}$, clamped at c. Further, Π_n is the set of all possible permutations of $\widehat{\mathbf{X}}$, and p is the order of the measure (often $p = 1$ or $p = 2$ are used). The benefit of the measure is that is captures the quality of both the cardinality estimate and the estimate of the state vectors of a multi-target filter. The drawback of the measure is that it is not as comprehensible as a simple RMSE measure.

To evaluate the performance of a tracking filter, we average over all state vectors, which gives us a definition of the *mean* OSPA (MOSPA) measure

$$\text{MOSPA}_p^{(c)}(\widehat{\mathbf{X}}) \triangleq \mathbb{E}_{p(\mathbf{X}|\mathbf{Z}^k)}\{\tilde{d}_p^{(c)}\}, \qquad (3.29)$$

which was first introduced in [67]. An optimal estimator, in the MOSPA sense, is an estimator which minimizes the MOSPA distance. Such an estimator is referred to as a minimum MOSPA (MMOSPA) estimator[2]. In Papers II, III and IV, MOSPA is used for the performance evaluations. Further, in [67], the calculation of MMOSPA estimates for known number of targets is described.

[2]Note that the relation between OSPA, MOSPA and MMOSPA is analogous to the relation between the common acronyms SE, MSE and MMSE.

Chapter 4

Radar Measurement Modeling

As described in Chapters 2 and 3, the purpose of a tracking algorithm is to recursively estimate quantities of interest, using information provided by one or several sensors. The quantities are often the kinematic states of objects, e.g., vehicles, whereas a sensor is a device that measures a physical quantity, and converts it into a signal that can be interpreted by a user or an instrument. An example of a sensor is a mercury thermometer, in which the temperature is measured by observing the expansion and contraction of a liquid. Two important features of a sensor are its *accuracy* and *resolution*, where the accuracy determine the trustworthiness of a measurement and the resolution is the smallest change the sensor can detect in the quantity it is measuring.

In target tracking, sensors that provide measurements of the range, direction and possibly speed of an object are preferred, since those are the states one is often interested in estimating. For short-distance applications, cameras are popular, while long-distance applications call for sensors that are less sensitive to light conditions and optical obstructions, such as clouds, and which are accurate at those distances. One sensor that fulfills those requirements is radar, which is the reason that it has been the most popular sensor in target tracking since its invention in the early 20th century. As seen in the previous chapters, the information from the sensor is incorporated in a Bayesian tracking framework through a sensor model. The model describes the density of a measurement, given the state of a target, and it provides information on the properties of the measurements; for example their accuracy. In this chapter, the basic measurement principles of radar are described, with the purpose of introducing the reader to the resolution modeling in Paper IV.

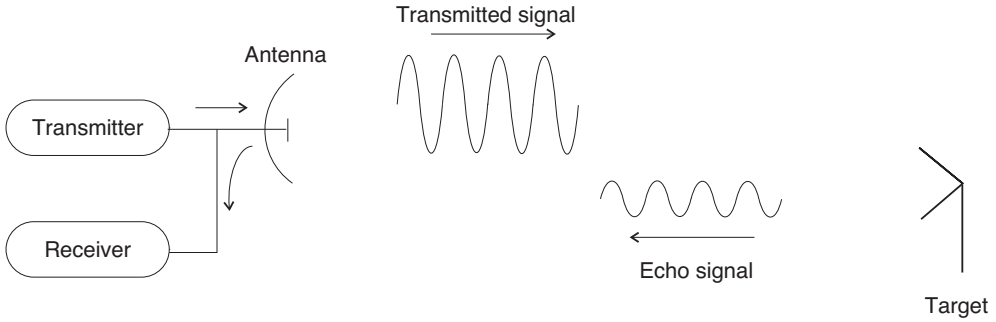

Figure 4.1: Basic principle of radar.

4.1 Basic Measurement Principles

In this section, the basic measurement principles of a detection radar [68] are described. More detailed descriptions are available in [68–71], and a good introductory textbook is [72]. The detection radars used in target tracking present measurements of one or several of the following properties: range, azimiuth angle, elevation angle and Doppler frequency (range rate). The accuracy and resolution in each of those dimensions are presented further on.

In its simplest form, as illustrated in Fig. 4.1, a pulsed radar transmits a single pulse and then awaits its response from a distant object. The radar then detects the received pulse, and measures the time, τ, it took for the pulse to travel to the target and back. The distance, or range, to the object is then estimated as

$$R = \frac{c\tau}{2}, \tag{4.1}$$

where c is the speed of light. The detection procedure is often carried out by a *matched filter* followed by a threshold detector, where the impulse response of a matched filter is the time-reversed and conjugated version of the transmitted pulse [70].

A pulsed radar often transmits pulses at regular intervals, and by observing the phase shift between pulses from the same object, the radar can estimate the Doppler shift of the target, and hence its speed. The fundamental relation between the radial speed, $\dot{R}$, and the Doppler shift, f_d, is

$$f_d = -\frac{2\dot{R}}{\lambda}, \tag{4.2}$$

where λ is the wavelength of the transmitted signal.

The azimuthal direction φ and the elevation angle θ to an object is determined either by the look direction of the radar, or by direction of arrival methods, such as monopulse [73], in which several antenna receive elements are used to determine the angular direction of an incoming electromagnetic wave.

The time interval between pulses is called pulse repetition interval (PRI), and its reciprocal is called pulse repetition frequency (PRF). Pulsed radar are classified by the PRF, and are divided into three groups: low PRF (LPRF), medium PRF (MPRF) and high PRF (HPRF). Each group has its strengths and weaknesses, and the selection of PRF depends on the application.

4.1.1 Radar Equation

The ability of a radar to detect the presence of an object mostly depends on the signal-to-noise ratio (SNR), where the signal part is the received radar reflection from an object, and where the noise is due to reflections from nature (called clutter) and thermal noise energy from the receiver and from external background noise. The SNR level in the radar is given by the radar equation, which is available in different forms, and where one form is

$$SNR\big(R,\theta,\varphi\big) = \frac{E_t G_t(\theta,\varphi) A_e(\theta,\varphi) \rho RCS}{(4\pi)^2 R^4 k T_0 C_B L_p}. \tag{4.3}$$

In (4.3), the parameters are defined as

E_t : energy in the transmitted signal waveform [Ws]

$G_t(\theta,\varphi)$: antenna gain of the transmitting antenna in (θ,φ) direction [ ]

$A_e(\theta,\varphi)$: effective receiving antenna aperture in (θ,φ) direction [m^2]

ρ : pulse-compression ratio (≤ 1) [ ]

RCS : radar cross section of the reflecting object [m^2]

kT_0C_B : noise energy [Ws]

L_p : additional propagation losses [ ].

The SNR has great impact on the tracking performance, since it determines the quality of the measurements being delivered to the tracking filter. It also determines how often the radar will see the different targets, which thus affects the update rate of a track. More directly, the SNR is used together with a detection probability curve [20] to determine the probability of detection, P_D, of the radar, under a given SNR (and given target signal fluctuation models, e.g., the family of Swerling models [74]).

4.1.2 Ambiguity

Consider the case of a pulsed radar transmitting a pulse every 1/PRF seconds. For the return from an object to be received in the interval between two pulse transmissions, the distance to the object cannot be larger than

$$R_{\max} = \frac{c}{2\text{PRF}}. \tag{4.4}$$

If the distance to an object is larger than $R_{\max}$, the system will not know if the reflection is from the first or the second pulse. We then have an *ambiguity* in range. To have unambiguous detection in the range dimension, we must hence have a PRF which is small—an LPRF system. However, in the Doppler domain, the ambiguousness is the opposite. In the Doppler processing, the radar performs a fast Fourier transform at a pulse-to-pulse basis, to find the Doppler shift of an object (at a certain range). The Doppler sample speed is thus equal to the PRF. Since the radar works with complex samples, the Nyquist theorem states that folding will occur at the complex Nyquist rate. Thus, the maximum unambiguous Doppler is

$$f_{\max} = \text{PRF}. \tag{4.5}$$

Therefore, to operate with unambiguous Doppler requires a system with a large PRF—a HPRF system. However, as seen above, a high PRF leads to range ambiguities. We thus have to make a trade-off between these ambiguities when designing a system. The ambiguity criterion in Doppler can be transformed into a range rate criterion, yielding the following maximum unambiguous relative speed of an object:

$$v_0 + v_{\max} = \frac{\text{PRF}\lambda}{2}. \tag{4.6}$$

In (4.6), v_0 is the speed of the sensor platform.

From the above description, we see that unambiguity in range leads to ambiguity in Doppler, and vice versa. Indeed, the definitions of the LPRF, MPRF and HPRF is actually such that an LPRF waveform is unambiguous in range, a HPRF waveform is unambiguous in Doppler, and an MPRF waveform is ambiguous in both dimensions. For more details of the applications of each type of waveform, and their advantages and disadvantages, see [70].

4.2 Accuracy

The accuracy of a radar sensor describes the precision with which the radar can detect an object in each measurement dimension. To specify the accuracy, a lower bound on the mean-square error is often provided, which states

the lowest possible variance of the error in the parameter of interest. Since many different lower bounds exist, the accuracy of a sensor is not specified by a single value, but depend on the bound and its tightness. Often, the Cramér-Rao bound [75] is used, but regardless of which bound that is used, the expressions have similar dependencies on, e.g., SNR and waveform characteristics.

In [71], the Cramér-Rao bound for time-of-arrival measurements of a radar is specified as

$$\sigma_\tau \geq \frac{1}{\beta\sqrt{2E/N_0}}, \tag{4.7}$$

where σ_τ is the standard deviation of the time-of-arrival error, and $\sqrt{2E/N_0}$ is the SNR after matched filtering. Converted to range measurements, the accuracy in range is determined by

$$\sigma_R \geq \frac{c}{2}\frac{1}{\beta\sqrt{2E/N_0}}, \tag{4.8}$$

where

$$\beta^2 = \frac{\int_{-\infty}^{+\infty}(2\pi f)^2\left|S(f)\right|^2 df}{\int_{-\infty}^{+\infty}\left|S(f)\right|^2 df} = \frac{1}{E}\int_{-\infty}^{+\infty}(2\pi f)^2\left|S(f)\right|^2 df \tag{4.9}$$

is the square of the effective bandwidth, or root mean square (RMS) bandwidth, of the waveform, and $S(f)$ is the Fourier transform of the transmitted signal $s(t)$. From (4.8), we see that the range accuracy is inversely proportional to the SNR and to the bandwidth. In [76], a tighter bound for time-of-arrival estimation is described, and the specific expression for a trapezoidal pulse-shape is derived.

By considering the connection between the spatial (angle) and spectral (frequency) domains, the angular accuracy of a radar is, according to [71], determined by

$$\sigma_\theta \geq \frac{1}{\gamma(\theta)\sqrt{2E/N_0}}, \tag{4.10}$$

where

$$\gamma(\theta)^2 = \frac{\int_{-\infty}^{\infty}(2\pi\theta/\lambda)^2\left|A(\theta)\right|^2 d\theta}{\int_{-\infty}^{+\infty}\left|A(\theta)\right|^2 d\theta} \tag{4.11}$$

is the squared effective aperture width of the antenna. In (4.11), $A(\theta)$ is the aperture illumination function, which describe the distribution of the current across the aperture (in the θ direction). For the angular accuracy in φ, the same equations as above are used, but with the aperture illumination $A(\varphi)$ in the φ dimension instead. The accuracy in angle is hence inversely proportional to the SNR and the aperture width of the antenna, where a wide aperture gives a narrow main lobe.

For the Doppler measurements, it has been shown that the accuracy is given by [77]

$$\sigma_{f_d} \geq \frac{1}{\alpha\sqrt{2E/N_0}}, \tag{4.12}$$

where

$$\alpha^2 = \frac{\int_{-\infty}^{+\infty}(2\pi t)^2 s^2(t)dt}{\int_{-\infty}^{+\infty} s^2(t)dt} \tag{4.13}$$

is the squared effective time duration of the transmit signal $s(t)$.

From the description in this section, we conclude that a high SNR accounts for more accurate measurements in all dimensions, and that the accuracy is further dependent on the waveform and antenna designs. Just as was observed in the ambiguity discussion, the range and the Doppler dimensions are connected also regarding accuracy. From (4.8) it is concluded that the range accuracy is inversely proportional to the bandwidth of the transmitted waveform, whereas the Doppler (and hence range rate) accuracy, according to (4.12), is inversely proportional to the effective time duration of the signal. Thus, for a simple pulse, high accuracy in both range and Doppler is not possible, since a long pulse corresponds to a small bandwidth. However, by using modulation techniques, it is possible to increase the bandwidth and thus obtain accuracy in both dimensions [71, 72]. Another consideration when designing a radar system is the measurement time. If high Doppler accuracy is required, a long pulse is necessary, which means that the search speed of the radar must be low. If a large area is to be covered, this will then affect the rate at which we can obtain measurements from the different targets, which in turn will affect the tracking performance.

4.3 Resolution

The concept of resolution determines how closely spaced two targets can be while still being distinguishable as two objects by the sensor. For range resolution, and the case of the transmit signal being a simple constant-frequency

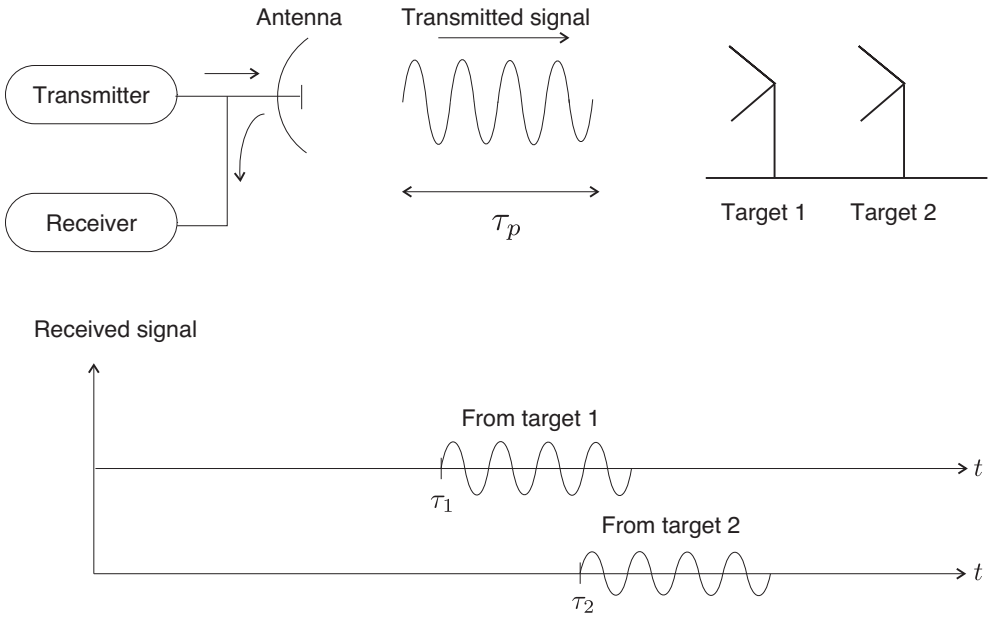

Figure 4.2: Ilustration of the range resolution for a simple pulse. For the two targets to be distinguishable by the radar, the time τ_2 must be larger than $\tau_1 + \tau_p$.

pulse of length τ_p, an illustration of the resolution concept is given in Fig. 4.2. For the two point reflectors to be distinguishable by the sensor, they must be separated with a distance

$$\Delta R = \frac{c\tau_p}{2}, \tag{4.14}$$

as the figure illustrates. This quantity, ΔR, is referred to as the range resolution of the radar. For more advances waveforms, the resolution does not simply depend on the pulse length, but actually depend on the bandwidth of the waveform, according to [68]

$$\Delta R = \frac{c}{2B}, \tag{4.15}$$

where B is the bandwidth of the baseband waveform, normally defined by the 3dB bandwidth. Note that, for a simple pulse, a reasonable approximation for the 3dB bandwidth is $B \approx 1/\tau_p$, which gives the expression in (4.14).

For the angular measurements in both azimuth and elevation, the resolution capability of the sensor depends on the antenna pattern [78]. For two closely spaced targets to produce two peaks in the received signal spectrum, they cannot be closer than the 3dB beamwidth of the mainlobe. Thus, the

angular resolution is

$$\Delta\theta = \theta_{3\text{dB}} \tag{4.16}$$
$$\Delta\phi = \phi_{3\text{dB}}, \tag{4.17}$$

where $\theta_{3\text{dB}}$ and $\phi_{3\text{dB}}$ are the 3dB beamwidths in the respective dimension. Note, however, that even though two targets are further apart than the angular resolution of the sensor, they may still not be resolvable, due to their radar cross sections being different—the electromagnetically smaller object might then drown in the reflection from the larger target.

In the Doppler domain, the resolution depends on the waveform of the transmitted pulse, and more specifically its spectrum. For a simple pulse of duration τ_p, the spectrum has a sinc shape, where the 3dB width of the main peak is $f_{3\text{dB}} = 1/\tau_p$. Thus, similar to the angular resolution, the resolution in Doppler is given by

$$\Delta f_d = f_{3\text{dB}} = \frac{1}{\tau_p}. \tag{4.18}$$

Noteably, by comparing with (4.14), good Doppler resolution is given by having a long observation time, i.e., a long pulse, whereas good range resolution is given by having a short pulse. The requirements are hence the opposite in range and Doppler. However, by modulating the pulse, a long pulse with large bandwidth can be created, whereby good resolutions in both dimensions are possible. This is the foundation for *pulse compression* [70]. Through the relation between Doppler frequency and range rate, the resolution in range rate is finally given by

$$\Delta\dot{R} = \frac{\Delta f_d \cdot \lambda}{2} = \frac{\lambda}{2\tau_p}. \tag{4.19}$$

The resolution properties of a waveform can be described through its *ambiguity function* [79], which describes the output of the matched filter as a function of the time delay and Doppler shift. It thus provides a description of how well the filter is matched to the received signal. By analyzing the ambiguity function, the expressions for resolution presented above can be found. However, for more precise analysis of the resolution capabilities, the signal-to-noise levels must be considered. A more rigorous analysis of the optimum range resolution in the presence of noise is given in [80].

When designing a radar system, the resolution, or resolution cell, is often chosen to match the size of the targets of interest. This is done in order to maximize the energy from the targets, to provide as high SNR as possible.

As long as all targets are of the same size, no resolution problems will occur, since the resolution is matched to the target size. However, when there are targets of different sizes, this approach is not possible. Then, if the resolution cell is matched to the largest targets, limited resolution problems may occur where several targets appear as one. On the other hand, if the cells are matched to the smallest targets, the larger targets (in this case called extended objects) will span over several resolution cells, and thus give rise to several measurements. In Paper IV, resolution problems of the first kind are discussed, and a resolution model is proposed which accounts for limited-resolution effects.

Chapter 5

Contributions and Future Work

In this chapter, the contributions of the thesis are summarized and a discussion of possible future work within the research area is held. The contributions take the form of four appended papers, which consider three different issues within targets tracking, namely multiple model filtering, data association and sensor resolution modeling. After the presentation of the contributions, an overview is given of possible directions of future work within the research area.

5.1 Publications

The thesis is based on the following publications:

Paper I *A New Multiple Model Filter with Switch Time Conditions*
The most common multiple model filter in target tracking is the Interacting Multiple Model (IMM) filter, which finds a sub-optimal solution to jump-Markov linear systems (JMLSs). An implicit assumption of JMLSs is that immediate model transitions have the highest probability. In this article, we argue that this model-shift property does not capture the typical nature of maneuvering targets, namely that changes in target dynamics usually persist for some time. We hence propose an adjusted switch time assumption that forces the dynamic models to remain fixed for a specified time. This is equivalent to replacing the Markov chain with a specific first-order semi-Markov chain. The modified filtering problem has lower complexity, and we derive a state estimation algorithm, called the Switch-Time Conditioned IMM (STC-IMM), which is close to optimal in many scenarios. Further, a deeper discussion of the relation between the proposed method and other

semi-Markov multiple-model methods is presented. A three-model version of the STC-IMM filter is compared with a three-model IMM filter on six benchmark problems for airborne target tracking. Results show an increase in performance for all six problems of the benchmark.

Paper II *Set JPDA Algorithm for Multi-Target Tracking*
In the development of data association and target tracking algorithms, the considered goal function has traditionally been the mean square error (MSE). Implicitly, this choice of goal function has a constraint that each target has an associated identity, or label. For many applications, this is in agreement with what the user wants, for example in surveillance systems where the long-term movement of specific targets are of interest. However, in other applications, such as radar cueing and collision avoidance, there is no interest in which target is which—the only interest is in where there are targets. To then use the MSE as cost function enforces an unnecessary constraint on the problem. A better choice is to use a measure which disregards target identity, and only considers the ability of the algorithm to estimate where there are targets. One such metric is the Mean Optimal Subpattern Assignment (MOSPA) metric, which we consider in this article. In this paper, we show how the traditional algorithms, which have been developed for the tracking problem with identity, can be adjusted to perform better when the identity is not of interest. More specifically, we propose an adjusted version of the Joint Probabilistic Data Association (JPDA) algorithm to this problem. When target identity is not an issue, the problem can be posed as the estimation of the random finite set (RFS) of the targets' states, through the calculation of the corresponding RFS density. However, the calculation of RFS densities is not tractable in general. The approach of this paper is instead to use the fact that many ordered (labeled) densities correspond to the same RFS density, and that the problem can be formulated as the finding of the best ordered density within the family of densities that correspond to the same RFS density. By this method, two extensions of the JPDA filter are presented: one by which optimal Gaussian approximations (which is used in the moment-matching step of the algorithm) in the Kullback-Leibler sense are found, and one which seeks to find both accurate Gaussian approximations, and better estimates. By evaluating the two filters, called Kullback-Leibler Set JPDA (KLSJPDA) and Set JPDA (SJPDA), on scenarios with closely-spaced targets—where the ordinary JPDA filter has problems—the benefits of the set-based approaches are showed. The results are compared with the JPDA filter and with the more complex Gaussian-Mixture Cardinalized Probability Hypothesis Density (GM-CPHD) filter, and they show that the set-based JPDA filters perform substantially better than JPDA, and almost

as well as the GM-CPHD filter.

Preliminary results have been published in [81].

Paper III *Performance Evaluation of MHT and CPHD in a Ground Target Tracking Scenario*

Comparisons between multi-target tracking algorithms are not common in the literature. In this paper, we evaluate two conceptually different tracking algorithms, namely a Multiple Hypothesis Tracking (MHT) algorithm and the recently-developed Gaussian-Mixture Cardinalized Probability Hypothesis Density (GM-CPHD) filter. As a reference, a conventional Single-Hypothesis Tracking (SHT) algorithm is applied to the scenario. To the best of our knowledge, a comparison between MHT and GM-CPHD has not been published before. The problem under consideration is tracking of nine densely spaced ground targets, using simulated measurements from an airborne radar. During parts of the scenario, up to two targets are not detectable by the radar, due to obscuration by a mountain. The scenario is challenging, which is indicated by the fact that SHT does not perform nearly as well as MHT and CPHD. Differences between the MHT and CPHD algorithms are pointed out and discussed. For example, GM-CPHD is more responsive to changes in the number of targets, whereas MHT is less responsive, but produces a more stable output.

Paper IV *Multitarget Sensor Resolution Model and Joint Probabilistic Data Association*

Traditionally, the design of target tracking algorithms has been performed under the assumption that all targets are always resolved by the sensor. However, when targets are closely spaced, this assumption does not hold, and it may lead to a degradation of the tracking performance due to an incorrect description of the data. In this paper, we present a framework for handling sensor resolution effects, and we propose a complete multitarget sensor resolution model that can be incorporated into traditional Bayesian tracking filters. Further, we derive the exact form of the posterior probability density function, and propose two alternative ways of approximating that density by a single Gaussian. The approaches can be viewed as two extensions of the Joint Probabilistic Data Association (JPDA) filter to resolution problems. The filters including a resolution model are evaluated on simulated radar data from a sensor with limited resolution, and the results are compared with the standard JPDA filter. As a measure of performance, the Mean Optimal Subpattern Assignment (MOSPA) measure is used. The results show increased tracking performance for the filters with a resolution model.

Preliminary results have been published in [82, 83].

5.2 Future Work

This thesis considers target tracking in complex scenarios. More specifically, it provides contributions within three areas of target tracking: multiple model filtering, data association and sensor resolution modeling. The common denominator between three of the appended papers (the fourth one being the performance evaluation) is that they identify weaknesses with existing methods for a range of problems, and propose alternative models or model structures that provide better descriptions of the data or the target dynamics. For the multiple model filtering work, the weakness is that it is not always reasonable that immediate model shifts are the most probable. Further, for the data association problem, it is noted that the traditional tracking algorithms can be improved when target identity is not of interest. Finally, the sensor resolution paper discusses the problem of limited resolution which, despite its recognized importance, has not been studied for more than two closely-spaced targets. Although the papers take a step forward in each direction, future work is needed within all these areas of research, and suggestions and discussions of such future work are presented here.

For the multiple-model filtering research, the area is rather well developed, and there is no clear view of what the next step is within that field. For the specific filter being presented in Paper I, however, there are more studies that would be interesting to pursue. First of all, evaluations of the filter performance on a wider set of scenarios where switch-time conditions are expected to be advantageous would be rewarding. Furthermore, the computational complexity of the filter and its correlation to the filter performance is a research study well worth considering. In Paper I, an approximation procedure referred to as early merging is proposed, and is expected to work well in many scenarios, due to the fast decay of the effect of prior information on the posterior density. However, a more rigorous study of when the approximation yields lower computational burden, while still performing better than a filter without switch-time conditions, is required.

The second paper of this thesis opens up an entirely new research area, and there are thus many potential research problems within that area. In Paper II, an adjustment of the traditional JPDA filter is presented. However, the conceptual method for improving any classic assignment algorithm is also presented in the paper. Natural future work is thus to consider more advanced data association algorithms, and to adjust them to the problem of target tracking without target identity. Since the most advanced algo-

rithm within the family of traditional algorithms is the MHT algorithm, an adjustment of that algorithm using the set-based methods of Paper II is an excellent proposal for future work. Further, the research so far has considered the number of targets to be known. In practice, that is rarely the case, and the extension to unknown number of targets is hence of utmost interest and importance.

Although being developed for the label-free problem, the SJDPA algorithm has the ability to state which target is which due to its foundation being the ordinary JPDA algorithm. However, in contrast to JPDA, the SJPDA algorithm can be developed to present the probability that a certain track belongs to a certain target. Let us try to understand why this may be valuable: if two targets are widely separated, the probability that one track belongs to target one, and the other to target two is 100%. But, when the targets move closer together, such that they share measurements in their respective validation gate, the tracks can no longer with certainty describe the one or the other target. Nevertheless, since the permutations of the SJPDA algorithm provide information on the amount of probability mass being shifted, it is possible to state the probability of a certain labeling. To develop such a procedure would widen the use of the SJPDA filter to problems where target identity is in fact important.

In Paper IV, a multi-target sensor resolution model is presented, and it is, to the best of the authors' knowledge, the first attempt at modeling the resolution limitations of a sensor for the case of more than two closely-spaced objects. There are hence many interesting research studies to perform in this area. One obvious continuation is to evaluate the resolution model on a wider set of scenarios, and also on real sensor data. Further, the implications need to be studied of the two alternative approaches to extend the JPDA filter to account for resolution limitations. More specifically, it is of interest to study the approximation accuracies of the posterior density for the two approaches, and their relative performance. An establishment of a lower bound on the estimation error under resolution problems is also an interesting subject for future work.

Just as the SJPDA algorithm, the sensor resolution model is developed for the situation when the number of targets is known. To extend the resolution model to unknown number of targets would be interesting, for example together with an MHT solution. An incorporation of a resolution model in FISST-based algorithms, such as the PHD and CPHD filters, is also a good candidate for future research. Since the JPDA filter, as most data association algorithms making soft decisions, experiences the problem of tracks being coalesced if they are close together, that is also a risk with the JPDA filters including a resolution model. Thus, to incorporate the resolution model into

a filter without track coalescence, such as the SJPDA paper, could be of high value.

Bibliography

[1] Christian P. Robert. *The Bayesian Choice*. Springer-Verlag, New York, 2001.

[2] G. Grimmett and D. Stirzaker. *Probability and Random Processes*. Oxford University Press, Oxford, third edition, 2001.

[3] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82 (Series D):35–45, 1960.

[4] A.H. Jazwinski. *Stochastic Processes and Filtering Theory*, chapter 8, pages 272–281. Academic, New York, 1970.

[5] S.J. Julier and J.K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Signal Processing, Sensor Fusion, and Target Recognition VI*, volume 3068 of *Proceedings of the SPIE*, pages 182–193, April 1997.

[6] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, 140(2):107–113, April 1993.

[7] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10:197–208, 2000.

[8] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter*. Artech House, London, 2004.

[9] Christos G. Cassandras and Stéphane Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic, Boston, USA, 1999.

[10] Fredrik Gustafsson. *Adaptive Filtering and Change Detection*. Wiley, Chichester, UK, 2000.

[11] H.A.P Blom. A sophisticated tracking algorithm for ATC surveillance radar data. In *Proceedings of the International Conference on Radar*, pages 393–398, May 1984.

[12] H.A.P Blom. An efficient filter for abruptly changing systems. In *Proceedings of the 23rd IEEE Conference on Decision and Control*, volume 23, pages 656–658, December 1984.

[13] H.A.P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(3):780–783, July 1988.

[14] V.S. Barbu and N. Limnios. *Semi-Markov Chains and Hidden Semi-Markov Models Toward Applications*. Lecture Notes in Statistics. Springer, New York, 2008.

[15] R.A. Howard. *Dynamic Probabilistic Systems*, volume II: Semi-Markov and Decision Processes. Wiley, New York, 1971.

[16] L. Campo, P. Mookerjee, and Y. Bar-Shalom. Failure detection via recursive estimation for a class of semi-Markov switching systems. In *Proceedings of the 27rd IEEE Conference on Decision and Control*, December 1988.

[17] L. Campo, P. Mookerjee, and Y. Bar-Shalom. State estimation for systems with sojourn-time-dependent Markov model switching. *IEEE Transactions on Automatic Control*, 36(2):238–243, February 1991.

[18] H.A.P. Blom. Hybrid state estimation for systems with semi-Markov switching coefficients. In *Proceedings of the European Control Conference*, July 1991.

[19] A.G. Zubov and A.I. Petrov. Estimation in nonlinear stochastic systems with sudden changes in the structure and state coordinates. *Soviet Journal of Computer and Systems Sciences (Tekhnicheskaya Kibernetika)*, 29(3):9–21, May-June 1991.

[20] Samuel Blackman and Robert Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, Norwood, MA, 1999.

[21] D.P. Bertsekas. The auction algorithm: a distributed relaxation method for the assignment problem. *Annals of Operations Research*, 14(1–4):105–123, June 1988.

[22] S. Coraluppi, C. Carthel, P. Willett, M. Dingboe, O. O'Neill, and T. Luginbuhl. The track repulsion effect in automatic tracking. In *Proceedings of the 12th International Conference on Information Fusion*, pages 2225–2230, July 2009.

[23] Y. Bar-Shalom and E. Tse. Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11(5):451–460, September 1975.

[24] T.E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8(3):173–183, July 1983.

[25] R. Fitzgerald. Track biases and coalescence with probabilistic data association. *IEEE Transactions on Aerospace and Electronic Systems*, 21(6):822–825, November 1985.

[26] H.A.P. Blom and E.A. Bloem. Probabilistic data association avoiding track coalescence. *IEEE Transactions on Automatic Control*, 45(2):247–259, February 2000.

[27] D.B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, December 1979.

[28] T. Kurien. Issues in the design of practical multitarget tracking algorithms. In Y. Bar-Shalom, editor, *Multitarget-Multisensor Tracking: Advanced Applications*. Artech House, Norwood, MA, 1990.

[29] G.C. Demos. Applications of MHT to dim moving targets. In *Proceedings of the SPIE - The International Society for Optical Engineering*, volume 1305, pages 297–309, April 1990.

[30] D.S. Chan, D.D. Harrison, and D.A. Langan. Tracking in a high-clutter environment: simulation results characterizing a bi-level MHT algorithm. In *Proceedings of the SPIE - The International Society for Optical Engineering*, volume 1954, pages 540–551, April 1993.

[31] D.S. Chan and D.A. Langan. Performance results of the bi-level MHT tracking algorithm for two crossing targets in a high clutter environment. In *Proceedings of the SPIE - The International Society for Optical Engineering*, volume 2235, pages 404–416, April 1994.

[32] I.J. Cox and S.L. Hingorani. An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):138–150, Feb. 1996.

[33] A.B. Poore and A.J. Robertson III. A new lagrangian relaxation based algorithm for a class of multidimensional assignment problems. *Computational Optimization and Applications*, 8(2):129–150, September 1997.

[34] S. Deb, M. Yeddanapudi, K. Pattipati, and Y. Bar-Shalom. A generalized s-d assignment algorithm for multisensor-multitarget state estimation. *IEEE Transactions on Aerospace and Electronic Systems*, 33(2):523–538, April 1997.

[35] S. Blackman. Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, 19(1):5–18, January 2004.

[36] T. Kurien. Report-to-target assignment in multisensor multitarget tracking. In *Proceedings of the 27th IEEE Conference on Decision and Control*, December 1988.

[37] K.G. Murty. An algorithm for ranking all the assignments in order of increasing cost. *Operations Research*, 16:682–687, 1968.

[38] R.W. Sittler. An optimal data association problem in surveillance theory. *IEEE Transactions on Military Electronics*, MIL-8(2):125–139, April 1964.

[39] I.R. Goodman, R.P. Mahler, and H.T. Nguyen. *Mathematics of Data Fusion*. Kluwer Academic Publishers, Boston, 1997.

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

[41] R. Mahler. Multitarget moment statistics and their application to multitarget tracking. In *Proceedings of the ONR Workshop on Tracking, in Honor of Yaakov Bar-Shalom*, Monterey, CA, USA, May 2001.

[42] L. Lin, Y. Bar-Shalom, and T. Kirubarajan. Track labeling and PHD filter for multitarget tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 42(3):778–795, July 2006.

[43] K. Panta, B-N. Vo, and D. Clark. An efficient track management scheme for the Gaussian-mixture probability hypothesis density tracker. In *Proceedings of the 4th International Conference on Intelligent Sensing and Information Processing*, pages 230–235, 2006.

[44] D.J. Daley. *An Introduction to the Theory of Point Processes*. Springer series in Statistics. Springer-Verlag, New York, 1988.

[45] R. Mahler. A theoretical foundation for the stein-winter probability hypothesis density (PHD) multitarget tracking approach. In *Proceedings of the 2000 MSS National Symposium on Sensor and Data Fusion*, volume I (unclassified), pages 99–117, San Antonio, TX, USA, June 2000.

[46] M.C. Stein and C.L. Winter. An additive theory of probabilistic evidence accrual. Technical Report LA-UR-93-3336, Los Alamos National Laboratories, Los Alamos, New Mexico, 1993.

[47] M.C. Stein and R.R. Tenney. What's the difference between PHS and MHT? working paper, Oasis Research and Alphatech, undated.

[48] R.P.S. Mahler. Multitarget Bayes filtering via first-order multitarget moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1152–1178, October 2003.

[49] P.R. Kalata. The tracking index: A generalized parameter for $\alpha - \beta$ and $\alpha - \beta - \gamma$ target trackers. *IEEE Transactions on Aerospace and Electronic Systems*, AES-20(2):174–182, March 1984.

[50] D. Svensson. *Multiple Model Filtering and Data Association with Application to Ground Target Tracking*. Licentiate thesis, Chalmers University of Technology, December 2008.

[51] R. Mahler. Phd filters of higher order in target number. *IEEE Transactions on Aerospace and Electronic Systems*, 43(4):1523–1543, October 2007.

[52] O. Erdinc, P. Willett, and Y. Bar-Shalom. Probability hypothesis density filter for multitarget multisensor tracking. In *Proceedings of the 8th International Conference on Information Fusion*, July 2005.

[53] H. Sidenbladh. Multi-target particle filtering for the probability hypothesis density. In *Proceedings of the 6th International Conference on Information Fusion*, pages 800–806, Cairns, Australia, July 2003.

[54] B-N. Vo and W.-K. Ma. The Gaussian mixture probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 54(11):4091–4104, November 2006.

[55] R. Mahler. A theory of PHD filters of higher order in target number. In *Proceedings of Signal Processing, Sensor Fusion and Target Recognition, XV*, Proceedings of the SPIE, Orlando, FL, USA, April 2006.

[56] O. Erdinc, P. Willett, and Y. Bar-Shalom. A physical-space approach for the probability hypothesis density and cardinalized probability hypothesis density filters. In *Proceedings of the SPIE - The International Society for Optical Engineering*, volume 6236, April 2006.

[57] D. Svensson and L. Svensson. An alternative derivation of the Gaussian mixture cardinalized probability hypothesis density filters. Technical Report R018/2008, Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden, November 2008.

[58] B.-T. Vo, B.-N. Vo, and A. Cantoni. Analytic implementations of the cardinalized probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 55(7):3553–3567, July 2007.

[59] B.-T. Vo, B.-N. Vo, and A. Cantoni. The cardinalized probability hypothesis density filter for linear Gaussian multi-target models. In *40th Annual Conference on Information Sciences and Systems*, March 2006.

[60] M. Ulmke, O. Erdinc, and P. Willett. Gaussian mixture cardinalized PHD filter for ground moving target tracking. In *Proceedings of the 10th international conference on information fusion*, July 2007.

[61] D.E. Clark, K. Panta, and B.-N. Vo. The GM-PHD filter multiple target tracker. In *Proceedings of the 9th International Conference on Information Fusion*, 2006.

[62] D. Fränken, M. Schmidt, and M. Ulmke. "Spooky action at a distance" in the Cardinalized Probability Hypothesis Density Filter. *IEEE Transactions on Aerospace and Electronic Systems*, 45(4):1657–1664, October 2009.

[63] J.R. Hoffman and R.P.S. Mahler. Multitarget miss distance via optimal assignment. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 34(3):327–336, May 2004.

[64] C.L. Givens and R.M. Shortt. A class of Wasserstein metrics for probability distributions. *Michigan Mathematical Journal*, 31(2):231–240, 1984.

[65] D. Schuhmacher, B.-T. Vo, and B.-N. Vo. On performance evaluation of multi-object filters. In *Proceedings of the 11th International Conference on Information Fusion*, June-July 2008.

[66] D. Schuhmacher, B.-T. Vo, and B.-N. Vo. A consistent metric for performance evaluation of multi-object filters. *IEEE Transaction on Signal Processing*, 56(8):3447–3457, August 2008.

[67] M. Guerriero, L. Svensson, D. Svensson, and P. Willett. Shooting two birds with two bullets: How to find minimum mean OSPA estimates. In *Proceedings of the 13th International Conference on Information Fusion*, July 2010.

[68] M.A. Richards. *Fundamentals of Radar Signal Processing*. McGraw-Hill, New York, 2005.

[69] D.K. Barton. *Modern Radar System Analysis*. Artech House, Norwood, MA, 1988.

[70] M.I. Skolnik. *Radar handbook*. McGraw-Hill, New York, 1990.

[71] M.I. Skolnik. *Introduction to Radar systems*. McGraw-Hill, New York, 2001.

[72] G.W. Stimson. *An Introduction to Airborne Radar*. SciTech Publishing, Inc., New Jersey, USA, 2nd edition, 1988.

[73] A.I. Leonov and K.I. Fomichev. *Monopulse Radar*. Artech House, Norwood, MA, 1986.

[74] P. Swerling. Probability of detection for fluctuating targets. *IRE Transactions on Information Theory*, 6(2):269–308, April 1960.

[75] S.M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*, volume I of *Prentice Hall Signal Processing Series*. Prentice Hall, New Jersey, USA, 1993.

[76] A.J. Weiss. Composite bound on arrival time estimation errors. *IEEE Transactions on Aerospace and Electronic Systems*, AES-22(6):751–756, 1986.

[77] R. Manasse. Summary of maximum theoretical accuracy of radar measurements. Technical Report MITRE Tech series Report 2, MITRE Corp., Bedford, MA, USA, 1960.

[78] C.A. Balanis. *Antenna theory*. Wiley, New York, 3rd edition, 2005.

[79] P.M. Woodward. *Probability and Information Theory with Applications to Radar*. McGraw-Hill, New York, 1953.

[80] N.J. Nilsson. On the optimum range resolution of radar signals in noise. *IRE Transactions on Information Theory*, 7(4):245–253, October 1961.

[81] L. Svensson, D. Svensson, and P. Willett. Set JPDA algorithm for tracking unordered sets of targets. In *Proceedings of the 12th International Conference on Information Fusion*, 2009.

[82] D. Svensson, M. Ulmke, and L. Danielsson. Multitarget sensor resolution model for arbitrary target numbers. In *Signal and Data Processing of Small Targets*, volume 7698 of *Proc. of SPIE*, 2010.

[83] D. Svensson, M. Ulmke, and L. Danielsson. Joint probabilistic data association filter for partially unresolved target groups. In *Proceedings of the 13th International Conference on Information Fusion*, July 2010.

Part II
Publications

Paper I

A New Multiple Model Filter with Switch Time Conditions

D. Svensson and L. Svensson

In *IEEE Transactions on Signal Processing*,
vol. 58, no. 1, January 2010.

A New Multiple Model Filter with Switch Time Conditions

Daniel Svensson* and Lennart Svensson†‡

Abstract

The interacting multiple model filter has long been the method of choice for performing target tracking using multiple motion models. The filter finds a sub-optimal solution to a problem that has the implicit assumption that immediate model shifts have the highest probability. When the sampling rate of the underlying continuous process is high compared to the target dynamics, this is not a reasonable assumption. Instead, changes in dynamics persist for some time. In this paper we propose an alternative switching model, which forces the dynamic models to persist for at least a model-specific time. The model is semi-Markov in nature, with a sojourn time probability mass function that is zero for a model-specific number of time steps, and then follows a geometrical distribution. Through this assumption a less complex problem in terms of model hypotheses arises, and to that problem we derive a state estimation algorithm that is close to optimal when the model assumptions are valid. Three other semi-Markov-based multiple-model filters are discussed and compared to in a qualitative sense. We also derive a new aircraft motion model for start and termination of turns. Finally, the proposed filter is evaluated on a benchmark scenario for tracking, and the results show a performance increase compared to IMM for the trajectories considered.

*D. Svensson is with the Department of Signals and Systems, Chalmers University of Technology, SE-412 96 Göteborg, Sweden (e-mail: daniel.svensson@chalmers.se).

†L. Svensson is with the Department of Signals and Systems, Chalmers University of Technology, SE-412 96 Göteborg, Sweden (e-mail: lennart.svensson@chalmers.se).

‡Manuscript received February 12, 2009; accepted July 16, 2009. This work was supported by the Swedish National Aeronautic Research Program (NFFP), which is funded by VINNOVA (The Swedish Governmental Agency for Innovation Systems). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. James Lam.

Index Terms

Benchmark problem, Interacting Multiple Model (IMM), Kalman filtering, multiple models, semi-Markov processes, state estimation, target tracking.

1 Introduction

This article is concerned with multiple-model estimation in a tracking framework. The main focus is on multiple motion models, which are incorporated to enable a good description of target dynamics. There are two main parts of the multiple-model framework, viz. the motion models themselves and the model for transition between dynamic models. In this article, which is an extension of a previous paper [1], we propose a new model structure for transitions between models, and for that structure we derive a close to optimal state estimation algorithm. We also present a new motion model for start and termination of turns.

The classical approach to multiple-model filtering is the jump Markov system (JMS), where target dynamics are captured by multiple motion models, and where the transition between those models is described by a first-order Markov chain. In the optimal solution to the problem, an exponentially growing number of model hypotheses need to be considered, which is infeasible in general. Sub-optimal solutions are found by pruning and/or merging branches of the hypothesis tree. Common merging algorithms are the Interacting Multiple Model (IMM) filter [2–4] and the Generalized Pseudo-Bayesian (GPB) algorithm [5–7]. An example of a pruning algorithm is the Multiple-Model Pruning (MMP) [8] algorithm.

In the Markovian assumption of JMS, it is implicitly assumed that immediate model shifts have the highest probability. The reason is that the duration time of a model, called sojourn time, has a probability mass function (pmf) that is geometrical for discrete-time Markov chains. In many scenarios, this is not in agreement with the actual target dynamics, in relation to the sampling rate of the underlying continuous-time process. Since the sampling rate is normally set equal to the measurement rate of the sensor, the Markovian assumption is not valid if the measurement rate is high compared to the target dynamics. Many typical radar systems have a time between measurements of around one to two seconds. If we want to use such a radar system to track aircraft with straight-ahead or turning motion, or off-road vehicles with stay-on-road or drive off-road motion, those maneuvers will often last for several seconds, which makes the Markovian assumption invalid. That targets rarely have the assumed property is well known, and it is stated in the survey article [9] that the geometrical distribution is "not

consistent with the duration of practical target motions".

To incorporate the sojourn-time dependence of target motion, the state vector can be augmented with the time spent in the model since the last model shift. By doing so, while keeping the Markovian property that the history of states does not affect the transition from the current to the next state, a semi-Markov chain [10] for state transition emerges. Classic work on discrete semi-Markov problems are, e.g., [11, 12]. Just as for jump-Markov systems, sub-optimal solutions are required to solve the jump semi-Markov problem. The natural first step is to adapt the IMM algorithm to semi-Markov chains, which has been performed in two similar ways. In [13] and [14] a sojourn-time dependent Markov IMM (STDM-IMM) algorithm is introduced, and in [15] a computationally less complex algorithm, called semi-Markov IMM (SM-IMM), is presented. These algorithms have the advantage of considering general sojourn time distributions, but to do so an immediate merging approximation is used, which limits the capability of incorporating the sojourn-time dependence. The algorithms are further based on an unfounded assumption, that generally does not hold for this class of problems, as is discussed in Section 3.2.2 below.

Another solution to jump semi-Markov state estimation is given in [16]. The algorithm is called latest change moment testing (LCMT), and is based on a change detection methodology. To reduce the number of hypotheses, the algorithm uses a sliding window pruning approach, where the length of the window determines the number of model shift hypotheses to consider. The algorithm also uses merging to further reduce the hypothesis tree. Positive aspects with LCMT is that it is a better approximation than SM-IMM and that it provides a maximum a posteriori (MAP) estimate of the time of the latest model change. Drawbacks include the increased computational complexity and the crude merging approximation.

When the measurement rate is high compared to the target dynamics, the probability of fast transitions is low. For problems of this kind, the approach of the current article is to approximate the small transition probabilities as zero, which makes the filtering problem simpler. The primary contributions of the article are twofold. First, we propose a new model structure for switching between multiple motion models. The structure has a switch time condition that forces the dynamic models to remain fixed for some time. In the sojourn time pmf, this is represented by an initial set of zeroes. To maintain a simple model, the zeroes are followed by a geometrical distribution. Since immediate transitions have zero probability, the problem is of less complexity than the traditional JMS problem in terms of number of model shift hypotheses. The second contribution of the article is the derivation of a close to optimal state estimation algorithm, for the considered family of problems.

Since the filter is derived for a specific model structure, it enables more accurate state estimation than the other semi-Markov-based algorithms, for problems captured by the new switching model. The proposed filter is evaluated on a benchmark tracking scenario [17] and compared to IMM. The analysis is a robustness analysis, in the sense that it is a comparison of two conceptually different filters applied to a problem that does not completely agree with any of the two filter assumptions. Results show a performance increase with the new filter for all six trajectories of the benchmark.

The article is structured as follows. In Section II, the state estimation problem is formulated. In Section III, classic state estimation methods and assumptions are discussed. The new approach is presented in Sections IV-V. Section VI is a short introduction to selection of design parameters for the new filter. In Section VII motion models for the considered benchmark scenario are discussed, and in section VIII the simulation results are presented. Section IX concludes the article.

2 Problem formulation

This article is about recursively approximating the posterior probability density function (pdf) $p\big(\mathbf{z}_k\big|\mathbf{Y}_{1:k}\big)$ using multiple motion models. The vector $\mathbf{z}_k$ is the state vector of a target at time index k and $\mathbf{Y}_{1:k} = \{\mathbf{y}_1\, \mathbf{y}_2\, \cdots \mathbf{y}_k\}$ is the set of all measurement vectors up to the current time instant k. The state vector is described by the partition

$$\mathbf{z}_k = \begin{bmatrix} \mathbf{x}_k^T & m_k & \mathbf{e}_k^T \end{bmatrix}^T. \tag{1}$$

The first two elements of the state vector are common to all multiple-model state estimation algorithms. The vector $\mathbf{x}_k$ represents the target's coordinates at time k and is called the target vector. The coordinates of the target is in a multi-dimensional coordinate system, which for example may contain position and velocity in Cartesian coordinates. The model variable $m_k \in \{1, 2, \ldots, M\}$ contains information on the currently active model, of the M possible ones. The third, possibly empty, element $\mathbf{e}_k$ of the state vector contains extra states that are required by some state estimation algorithms, like the one presented in this article.

Motion models describe the evolution of the state vector over time. To capture the behavior of real targets, multiple motion models are often needed. The state estimation algorithm presented in this paper is applicable to both nonlinear and linear models. A general, nonlinear, motion model is described by

$$\mathbf{x}_{k+1} = f\left(\mathbf{x}_k, m_k, \mathbf{v}_k(m_k)\right), \tag{2}$$

where f is a nonlinear (prediction) function and $\mathbf{v}_k(m_k)$ is a realization of the model-dependent process noise, which has some assumed distribution. The corresponding linear motion model is written as

$$\mathbf{x}_{k+1} = \mathbf{A}(m_k)\mathbf{x}_k + \mathbf{B}(m_k)\mathbf{v}_k(m_k). \tag{3}$$

To complete the multiple motion model, we need a switching model that describes the evolution of m_k over time. The switching model is described by the transition probabilities $\pi_{ij}(k)$ from model i to model j at transition times k.

The target vector $\mathbf{x}_k$ is related to the measurement vector $\mathbf{y}_k$ through the measurement model. For the nonlinear case it is defined as

$$\mathbf{y}_k = h\left(\mathbf{x}_k, \mathbf{w}_k\right), \tag{4}$$

where h is a nonlinear function and $\mathbf{w}_k$ is measurement noise with some assumed distribution. In the linear case, the corresponding model is given by

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{w}_k. \tag{5}$$

The model parameter m_k is omitted in these measurement model descriptions, since treatment of multiple measurement models is not the focus of this article. Finally, $\mathbf{v}_k$ and $\mathbf{w}_k$ are white and independent.

3 Classical state estimation methods and assumptions

3.1 Markov-based methods

When target dynamics are modeled with a multiple-model approach, it is most commonly done with the jump Markov system assumption. This means that the target dynamics is described by a set of motion models, and that the dynamics at a given time instant is captured by one of those models. Furthermore, the active model can change over time, and this model shift is described by a first-order Markov chain. The transition probability of the chain at time index k is defined as

$$\pi_{ij}(k) = P\{m_{k+1} = j \big| m_k = i\}. \tag{6}$$

By the Markovian assumption, the following holds true for the model switching

$$P\{m_{k+1}=j \big| m_k=i, T_k=s\} = P\{m_{k+1} = j \big| m_k = i\}, \quad \forall s \in \mathbb{Z}^+, \tag{7}$$

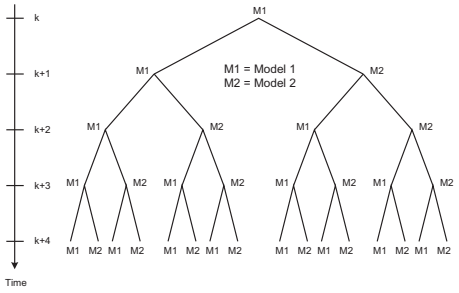

Figure 1: Model hypothesis tree for two-model jump-Markov systems, where it is assumed that model M1 is valid at time k.

where T_k represents the duration of model i at time k. Thus, there is a memoryless property of the dynamics, which implies that the probability of having a model shift at a certain time is independent of the time when the current model became active. The Markovian assumption also implies that the sojourn time is exponentially distributed for continuous-time models and geometrically distributed for discrete-time models. Due to this implicit assumption, immediate transitions are assumed to have the highest probability, which is a non-intuitive property for most targets. We will discuss more about these properties and their validity for maneuvering targets further on.

3.1.1 Conceptual solution

Consider the model hypothesis tree in Fig. 1. At time instant k, each path through the hypothesis tree represents a plausible sequence of models (called regimes), denoted

$$r^l_{1:k} = \left\{ r^l_1, r^l_2, \ldots, r^l_k \right\}, \quad l = 1, 2, \ldots, R_k. \tag{8}$$

The value of r^l_i determines the active model in the time interval $[i, i+1)$, of the l^{th} regime, and the variable R_k contains the total number of regimes at time k. For an M-model scenario, the number of regimes grows as M^k. In the conceptual solution to the problem, all these hypotheses must be considered. That is, the posterior density takes the form of a mixture[1]

$$p(\mathbf{x}_k \big| \mathbf{Y}_{1:k}) = \sum_{l=1}^{R_k} p(\mathbf{x}_k \big| r^l_{1:k}, \mathbf{Y}_{1:k}) P\{r^l_{1:k} \big| \mathbf{Y}_{1:k}\}. \tag{9}$$

Due to the rapid increase in number of hypotheses, sub-optimal solutions are required. The main tools for the common sub-optimal solutions are

[1] A Gaussian mixture for linear Gaussian models.

pruning and merging, where pruning resembles the removal of low-probability branches, and merging is a unification of similar branches into a single one.

3.1.2 IMM and its approximations

By far the most popular sub-optimal solution to the Markovian switching problem is the Interacting Multiple Model (IMM) filter [2–4]. At each time instant, the filter approximates the posterior density by a Gaussian mixture, with one Gaussian pdf connected to each model. To understand the approximations of the filter, again consider the model hypothesis tree of Fig. 1. To reduce the number of hypotheses, the approach of the IMM filter is to merge all branches that belong to the same model at each time instant. Thus, the filter has a bank of M model-matched filters, where M is equal to the number of models. At each time instant, the input to each filter is a Gaussian approximation of a mixture of the M model-matched filter pdfs of the previous time instant. The mixing probability depends on the model probability at the previous time and the transition probability matrix $\mathbf{\Pi}$ of the Markov chain. Since the proposed filter has similarities to IMM, the IMM equations are described for reference.

The probability of model i at time $k-1$ is defined as

$$w_{k-1}^{i} = P\{m_{k-1} = i \big| \mathbf{Y}_{1:k-1}\}, \tag{10}$$

and the mixing probability for model j, given transition from model i, as

$$\mu_{k-1}^{i|j} = P\{m_{k-1} = i \big| m_k = j, \mathbf{Y}_{1:k-1}\}. \tag{11}$$

Using Bayes' formula, this is rewritten as

$$\mu_{k-1}^{i|j} = \frac{\pi_{ij} w_{k-1}^{i}}{\sum_{i=1}^{M} \pi_{ij} w_{k-1}^{i}} = \frac{\pi_{ij} w_{k-1}^{i}}{w_{(k-1)}^{*j}}, \tag{12}$$

where π_{ij} is the transition probability, defined in (6), M is the number of models, and $w_{(k-1)}^{*j}$ is the weight of model j after mixing. The input to each model-matched filter j is hence a Gaussian approximation represented by its mean $\mathbf{x}_{k-1|k-1}^{(j)}$ and covariance matrix $\mathbf{P}_{k-1|k-1}^{(j)}$, given by

$$\mathbf{x}_{k-1|k-1}^{(j)} = \sum_{i=1}^{M} \mu_{k-1}^{i|j} \hat{\mathbf{x}}_{k-1|k-1}^{i} \tag{13}$$

$$\begin{aligned} \mathbf{P}_{k-1|k-1}^{(j)} = \sum_{i=1}^{M} \mu_{k-1}^{i|j} \Big[& \mathbf{P}_{k-1|k-1}^{i} \\ & + \left(\hat{\mathbf{x}}_{k-1|k-1}^{i} - \mathbf{x}_{k-1|k-1}^{(j)}\right)\left(\hat{\mathbf{x}}_{k-1|k-1}^{i} - \mathbf{x}_{k-1|k-1}^{(j)}\right)^{T} \Big]. \end{aligned} \tag{14}$$

After the model-matched filters have been applied, the output of each filter is the posterior density $p(\mathbf{x}_k|\mathbf{Y}_{1:k}, m_k = j) = \mathcal{N}\left(\mathbf{x}_k;\, \mathbf{x}^j_{k|k}, \mathbf{P}^j_{k|k}\right)$ and the respective model probability is updated as

$$w^j_k = P\{m_k = j|\mathbf{Y}_{1:k}\} = \frac{\Lambda^j_k w^{*j}_{(k-1)}}{\sum_{i=1}^{M} \Lambda^i_k w^{*i}_{(k-1)}}, \tag{15}$$

where Λ^j_k is the model-conditioned likelihood function,

$$\Lambda^j_k = p\big(\mathbf{y}_k|\mathbf{Y}_{1:k-1}, m_k = j\big) = \mathcal{N}\left(\boldsymbol{\nu}^j_k; \mathbf{0}, \mathbf{S}^j_k\right). \tag{16}$$

In (16), $\boldsymbol{\nu}^j_k$ and $\mathbf{S}^j_k$ are the innovation and the innovation covariance matrix, respectively, obtained by the j^{th} model-matched filter. Finally, the posterior density is approximated by a Gaussian approximation of the mixture

$$p\big(\mathbf{x}_k|\mathbf{Y}_{1:k}\big) \cong \sum_{i=1}^{M} w^i_k\, p\big(\mathbf{x}_k|m_k = i, \mathbf{Y}_{1:k}\big). \tag{17}$$

3.2 Semi-Markov methods

With the semi-Markov methodology, the Markov chain of the model transitions is replaced with a semi-Markov chain [10], for which the transition probabilities depend on the sojourn time. For exponential/geometrical sojourn-time assumptions, the semi-Markov chain reduces to an ordinary Markov chain. The transition probability from state i to j at time k for a discrete-time semi-Markov chain is defined as

$$\pi^{T_k}_{ij}(k) = P\{m_{k+1} = j|m_k = i, T_k\} \tag{18}$$

where T_k represents how long model i has been active since the last model shift. Defining the sojourn time $\tau_i(k)$ of model i at time k as

$$\tau_i(k) = t_j - t_i, \tag{19}$$

where t_i is the time of transition to model i and where t_j is the time of transition from model i to model j, for the current period of model i, it holds that $T_k = \tau_i(k)$ when the shift to model j occurs. At the time of transition, a semi-Markov chain behaves in a Markovian manner, meaning that the history of states does not affect the transition probability to any new state — only the current state plays a role in that sense. To describe the sojourn time dependence, the state vector includes T_k as an extra state, i.e., $\mathbf{e}_k = e_k = T_k$ in (1).

3.2.1 Conceptual solution

In the optimal solution to the semi-Markov switching multiple-model problem, we are interested in describing the posterior density $p\big(\mathbf{x}_k\big|\mathbf{Y}_{1:k}\big)$. The expression for this density is a mixture of densities, equivalent to the conceptual solution of Markov-based methods in (9). The difference between the solutions for Markov and semi-Markov systems lies in the calculation of the regime history probability $P\left\{r_{1:k}^l\middle|\mathbf{Y}_{1:k}\right\}$ (for each regime history l), which is a consequence of the transition probabilities being sojourn-time dependent for semi-Markov chains. As the number of hypotheses, or regime histories, grows exponentially over time, the conceptual solution is in general infeasible.

3.2.2 Sub-optimal solutions

Several attempts have been made to approximate the posterior pdf in semi-Markov switching problems. In [13] and [14] the sojourn-time dependent Markov (STDM) IMM algorithm was presented, and in [15] a computationally less complex algorithm was described, called semi-Markov IMM (SM-IMM). Since the latter algorithm requires less computations with the same performance [15], we focus on that one (both algorithms are based on similar ideas).

Just as for IMM, the SM-IMM algorithm requires M model-matched filters. The difference is that the transition probabilities are sojourn-time dependent and that they change over time. The filter requires knowledge of the model-dependent sojourn-time pmf $P\left\{\tau_i(k)\middle|\text{Transition from model } j\right\}$, $i,j \in \{1,\ldots,M\}, i \neq j$, which can have arbitrary form. At each time instant, the sojourn time of each model is estimated in a recursive manner, and the transition probabilities at time k are then calculated from prior knowledge (the sojourn time pmf) and the likelihood of the sojourn time, given model and data up to the current time. In order to obtain the expressions for calculating the transition probabilities, the following assumption is made

$$p(\mathbf{x}_k\big|m_k,\mathbf{Y}_{1:k},T_k) = p(\mathbf{x}_k\big|m_k,\mathbf{Y}_{1:k}), \tag{20}$$

i.e., that $\mathbf{x}_k$ is conditionally independent of the sojourn time, given the active model at time k and the measurements up to time k. This assumption is only valid if the system is Markovian, i.e., if model changes follow a Markov chain. For semi-Markov chains it is hence not true, so the SM-IMM approach is only valid if the system is close to Markov. Further, (20) implies that

$$P\left\{T_k\middle|m_k,\mathbf{Y}_{1:k}\right\} = P\left\{T_k\middle|m_k,\mathbf{Y}_{1:k-1}\right\}, \tag{21}$$

that is that the latest measurement does not affect the probability that model m_k has been valid for T_k seconds, which is non-intuitive. Actually, in the derivation of the STDM-IMM algorithm, (21) is erroneously reasoned to always hold true — a statement proven false in [18].

The advantage with the SM-IMM algorithm is that it can handle arbitrary sojourn-time distributions. The disadvantages with the approach are that it is a rather crude approximation, due to the immediate merging of posterior densities conditioned on different sojourn times, and that it is based on the above unfounded assumption. A further important difficulty is that it is not clear how to select the sojourn-time dependent transition probabilities $\pi_{ij}^{T_k}$.

Another solution to jump semi-Markov state estimation is given in [16]. The algorithm is called latest change moment testing (LCMT), and is based on a change detection methodology. The algorithm was argued for in [18], where a comparative study with STDM-IMM showed significantly better performance in a scenario with semi-Markov switching, at the cost of higher computational burden. The output of the estimation filter is first the MMSE estimate of the state vector and its covariance matrix, and secondly the MAP estimate of the time of the latest model change.

In the LCMT setting, a hypothesis is the time of the latest model change. An example of a hypothesis is hence that the latest shift to model j occurred at time $k-2$, and that the model has been valid thereafter. The algorithm has a deferred decision logic of v time steps, which means that a hypothesis is not ruled out until v time steps have passed. That is, if $v = 3$ the hypothesis that there was a shift to model j at time $k-3$, is kept at least until time k. In order to reduce the number of hypotheses, a sliding window pruning approach is used. Only the best hypothesis v steps back in time is kept, which implies that M hypotheses are pruned in each iteration.

At each time instant of the LCMT algorithm, M new hypotheses are spawned, where each hypothesis represents a sudden change of models to model $j \in \{1, 2, \ldots, M\}$. Since M hypotheses are pruned at each time instant, the total number

$$J = Mv + 1 \tag{22}$$

of filters are kept constant. Each new hypothesis is a merging of J hypotheses, since transition occurs from each of the hypotheses at the previous time. The transition probabilities are model and sojourn-time dependent. For each hypothesis, a model-matched filter is run.

The benefits with the algorithm are that it can handle general sojourn-time pmfs in a less approximate way than the semi-Markov IMM algorithms, and that a MAP estimate of the latest model change is given, which is of

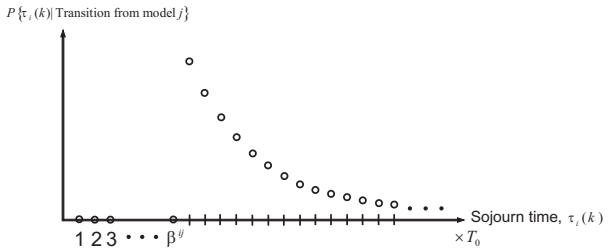

Figure 2: Illustration of the proposed probability mass function of the model sojourn time, $P\left\{\tau_i(k)\middle|\text{Transition from model } j\right\}$. The pmf is zero up to the model-dependent time $T_0\beta^{ij}$, and then follows a geometrical distribution.

interest in some applications regarding detection of failure. The drawback is the need for more hypothesis-matched filters, and the difficulty to select the sojourn-time dependent transition probabilities. The LCMT algorithm, as well as the other semi-Markov algorithms, is directed towards the general semi-Markov problem. In this article, however, we propose a new model structure, which is less general but that is applicable to a wide set of problems. For that model structure, the referred semi-Markov algorithms all provide a worse approximation than the proposed filter.

4 The new approach

In situations where the target dynamics are low compared to the sampling rate of the underlying continuous-time process, the probability of early transitions is low. The jump-Markov methods, presented in the previous section, are then limited by the assumption that immediate transitions have the highest probability. The referred semi-Markov methods, on the other hand, have the possibility of incorporating general distributions. However, in order to do so, they introduce limiting merging approximations. In this section we propose a new model structure with sojourn-time conditioned switching. Through this model a less complex problem in terms of model hypotheses arises, and to that problem we derive a state estimation algorithm, called the switch-time-conditioned IMM (STC-IMM) filter, which is close to optimal when the model assumptions are valid.

4.1 Model structure

The idea with the proposed model structure is to have a switch time condition that forces the models to remain fixed for some time. In the sojourn time

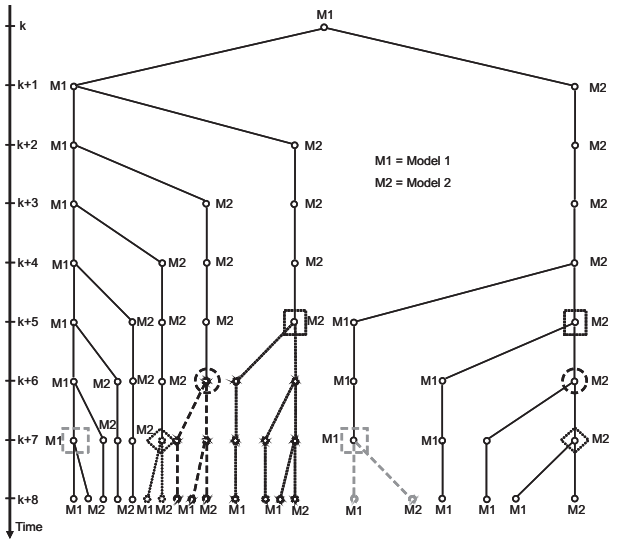

Figure 3: Model hypothesis tree for a two-model case with $\beta^{12} = 2$ and $\beta^{21} = 3$. Here it is assumed that model 1 is valid at time k, and has been so for more than β^{12} time steps. At time step $k + 8$ there are 20 branches in the tree. With the jump Markov approach, the number of branches would be 256, which is an indication of the much slower tree growth yielded by the approach of this paper. The merging approximations introduced by the proposed algorithm are also marked, where nodes to merge are enframed by a square, a circle or a diamond. Dashed branches represent hypotheses that vanish when nodes are merged. For the proposed algorithm, the number of modes would be reduced from 20 to 7 (cf. (27)) in this scenario.

pmf, this is represented by an initial set of zeroes, as illustrated in Fig. 2. For transition from model j to model i, the number of zeroes is β^{ij}. By studying the accompanying model hypothesis tree, illustrated in Fig. 3, we see that the tree growth is slower than for jump-Markov systems (cf. Fig. 1). From a model hypothesis point of view, we hence obtain a less complex problem, which at the same time is more realistic in situations where the sampling rate is high compared to the dynamics. The new model structure introduces a reasonable approximation if the true pmf has close to zero probability of transition for the first β^{ij} time instants. As was shown in [1], the depth can be kept small in many situations, since the gain in having a filter depth is most significant for small β^{ij}. For simplicity, after the initial set of zeroes, the pmf is assumed to follow a geometrical distribution, which is parameterized by the model-specific mean survival time τ_S^{ij}.

4.2 Conceptual solution

Consider the model hypothesis tree of Fig. 3. In the optimal solution of the state estimation, we must consider all hypotheses of the tree, i.e., in the description of the posterior pdf $p\big(\mathbf{x}_k|\mathbf{Y}_{1:k}\big)$, we must add the contributions from each and every branch of the tree. In the same way as for the Markov methods, we define a regime as

$$r_{1:k}^l = \left\{r_1^l, r_2^l, \ldots, r_k^l\right\}, \quad l = 1, 2, \ldots, S_k, \tag{23}$$

where S_k is the total number of regimes at time k. In general we have that (cf. (8))

$$S_k \leq R_k \tag{24}$$

where equality holds only for the special case of $\beta^{ij} = 0$; $i, j \in \big\{1, \ldots, M\big\}$. The posterior density at time k is the mixture

$$p\big(\mathbf{x}_k\big|\mathbf{Y}_{1:k}\big) = \sum_{l=1}^{S_k} p\big(\mathbf{x}_k\big|r_{1:k}^l, \mathbf{Y}_{1:k}\big) P\big\{r_{1:k}^l\big|\mathbf{Y}_{1:k}\big\}. \tag{25}$$

Since S_k is smaller than R_k in all cases considered here, the upper limit of the summation grows more slowly than for the jump-Markov systems, which means that larger tree depths can be considered. However, the number of regimes is ever growing even for this model structure, so a sub-optimal solution is required.

4.3 Sub-optimal solution: the STC-IMM filter

Again consider Fig. 3. As we can see, several branches of the tree have equal regime histories for the last consecutive time steps. This means that if a model-matched filter is run for each regime, several filters have performed state prediction using the same process model, and measurement update with the same measurements[2], for the last consecutive time steps. From Kalman filter theory it is known that the influence of prior information decays exponentially with time. Thus, the posterior densities of the model-matched filters with equal regimes for the last time steps are similar. A good approximation is hence to merge these similar branches of the tree. Consequently, we introduce the following approximation

$$\begin{aligned} &p(\mathbf{x}_k\big|\mathbf{Y}_{1:k}, m_k, T_k > T_0\beta^{ij}) \\ &\qquad \cong p(\mathbf{x}_k\big|\mathbf{Y}_{1:k}, m_k, T_k = T_0\beta^{ij}), \end{aligned} \tag{26}$$

[2]Note that data association problems are not considered in this article.

i.e., knowledge regarding the active model more than β^{ij} time steps back is uninformative, given that we know that the current model, m_k, has been active ever since. The merging time of the new filter is thus set to the time instant before model shifts are first possible, i.e., at time instant β^{ij}. The merged nodes of the hypothesis tree are marked in Fig. 3. Only one filter and one probability weight are required for each merged node, due to the geometrical distribution in the tail of the sojourn time pmf. According to [19], a typical rule of thumb is that merging can be performed, with negligible effect on filter performance, when the model-matched filters have been updated with the same measurements for at least the last three time steps. Thus, according to this rule, any value of β^{ij} larger than two implies that the merging approximation is reasonable, which means that a close to optimal solution is obtained.

The proposed multiple-model state estimation filter is built up by a set of model-matched filters. These are not only associated with a motion model, but also to the time spent in each model and the model from which the transition occurred. In Fig. 4, an example of the mode set for model 1 in a three-model STC-IMM filter is shown. The top three white circles represent the hypotheses that the transition to model 1 was from model 2, and that model 1 has been valid for 1, 2 and $\beta^{12} = 3$ time steps, respectively. The bottom four white circles represent transition from model 3 and the hypotheses that model 1 has been valid for 1 to $\beta^{13} = 4$ time steps, respectively. These hypotheses are called *static* modes. Transition from the model is only possible from the *transition* mode, which is the black circle in Fig. 4. Each model thus comprises a set of static modes and one transition mode. The total number of modes (model-matched filters) are hence

$$N = M + \sum_{i=1}^{M} \sum_{\substack{j=1 \\ j \neq i}}^{M} \beta^{ij}, \tag{27}$$

where M is the number of models as well as the number of transition modes.

4.3.1 State estimation algorithm

The STC-IMM filter has four algorithmic steps: transition mode mixing, mode transfer, filtering and MMSE estimation. The notation used to describe the algorithm is the following. For model $i \in \{1, \ldots, M\}$, the static modes are represented by s_{ij}^{l}, $l = 1, \ldots, \beta^{ij}$, $j \neq i$, and the transition mode by T_i. The mode weights are denoted $w_k^{s_{ij}^{l}}$ for the static modes and $w_k^{T_i}$ for the transition modes. The rightmost static modes (white circles) in Fig. 4 are

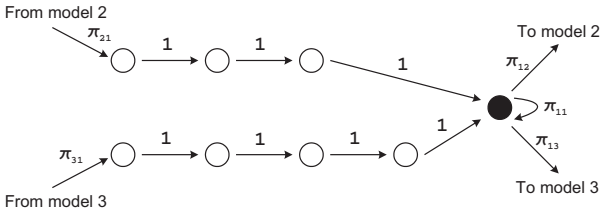

Figure 4: Illustration of the mode set for model 1 in a three-model STC-IMM filter with $\beta^{12} = 3$ and $\beta^{13} = 4$. The white circles represent static modes, and the black circle the transition mode of the model. The arrows indicate transitions that occur at each iteration, and the number or variable (π) close to the arrow represent the probabilities with which they occur.

called the highest-order static modes, and are represented by $s_{ij}^{\beta^{ij}}$. A pseudo code algorithm description is found in Table 1. Note that the table contains additional features of the algorithm, which are described in Section 5. In the following, we describe the algorithmic steps in more detail.

Step I: Transition mode merging

The first step of the algorithm is the realization of the approximation described in (26). It is the merging of the highest-order static modes, $s_{ij}^{\beta^{ij}}$, and the transition mode, T_i, of each model i, represented by the rightmost white circles and the black circle in Fig. 4, respectively. The merging is performed in an IMM fashion, where the weighted sum of M Gaussian pdfs are approximated by a single Gaussian pdf. The result is used to approximate the posterior density of the transition mode.

In the equations that follow, each mode is represented by a Gaussian pdf, and we assume that all $\beta^{ij} \neq 0$, i.e., that we have a full collection of static modes[3]. For the transition mode, T_i, of model i, the posterior density at time k is described by the mean value $\hat{\mathbf{x}}_{k|k}^{T_i}$ and the covariance matrix $\mathbf{P}_{k|k}^{T_i}$. The highest-order static modes $s_{ij}^{\beta^{ij}}$ of model i are described by the mean values $\hat{\mathbf{x}}_{k|k}^{s_{ij}^{\beta^{ij}}}$ and covariance matrices $\mathbf{P}_{k|k}^{s_{ij}^{\beta^{ij}}}$ for $j = 1, \ldots, M; j \neq i$. The transition

[3]In situations where $\beta^{ij} = 0$ for transition to model i from some models j, there are no static modes for those transitions. In the merging, the (non-existing) highest-order static modes are replaced by the corresponding transition modes of models j. Note that if $\beta^{ij} = 0, \forall i, j$, the STC-IMM filter is equivalent to the IMM filter.

Table 1: STC-IMM algorithm description

I: Transition mode mixing

for each model $i \in M$, do

if $\beta^{ij} \neq 0,\ j = 1, \ldots, M;\ j \neq i$

mix highest-order static modes (with $w_k^{s_{ij}^{\beta^{ij}}} > \gamma$) with transition mode (if $w_k^{T_i} > \gamma$), see (28)–(31)

else

mix transition mode with highest-order static modes (for $\beta^{ij} \neq 0$), and with transition mode of model j (if $\beta^{ij} = 0$) (and use only the modes with weights $> \gamma$)

II: Mode transfer

for each model $i \in M$, do

transfer probability mass from transition mode T_i to first-order static modes s_{ji}^1 of models $j \neq i$ (with transition probability π_{ij}), transfer probability mass from each static mode s_{ij}^l in the mode sets of model i to the one step higher order static mode s_{ij}^{l+1}, $l = 1, \ldots, \beta^{ij} - 1$ (with transition probability 1)

III: Filtering

for each static and transition mode, do

if mode weight $w_k^{s_{ij}^l} > \gamma \ / \ w_k^{T_i} > \gamma$

perform filtering using linear or nonlinear filter (Kalman filter, EKF, UKF, ...)

else

set $w_k^{s_{ij}^l} = 0 \ / \ w_k^{T_i} = 0$, and perform no filtering

IV: MMSE estimation

Compute $\hat{\mathbf{x}}^{\mathrm{MMSE}}$, see (32)

mode merging is thus described by (cf. (13)–(14))

$$\mathbf{x}_{k|k}^{(T_i)} = \mu_k^{T_i}\hat{\mathbf{x}}_{k|k}^{T_i} + \sum_{\substack{j=1 \\ j\neq i}}^{M} \mu_k^{j|i}\hat{\mathbf{x}}_{k|k}^{s_{ij}^{\beta ij}} \tag{28}$$

$$\begin{aligned}\mathbf{P}_{k|k}^{(T_i)} = \mu_k^{T_i}\Bigg[\mathbf{P}_{k|k}^{T_i} + \left(\hat{\mathbf{x}}_{k|k}^{T_i} - \mathbf{x}_{k|k}^{(T_i)}\right)\left(\hat{\mathbf{x}}_{k|k}^{T_i} - \mathbf{x}_{k|k}^{(T_i)}\right)^T\Bigg] \\ + \sum_{\substack{j=1 \\ j\neq i}}^{M} \mu_k^{j|i}\Bigg[\mathbf{P}_{k|k}^{s_{ij}^{\beta ij}} + \left(\hat{\mathbf{x}}_{k|k}^{s_{ij}^{\beta ij}} - \mathbf{x}_{k|k}^{(T_i)}\right) \\ \times \left(\hat{\mathbf{x}}_{k|k}^{s_{ij}^{\beta ij}} - \mathbf{x}_{k|k}^{(T_i)}\right)^T\Bigg],\end{aligned} \tag{29}$$

where (T_i) denotes the transition mode after mixing. The mode mixing probability $\mu_k^{T_i}$ for the transition mode is

$$\mu_k^{T_i} = \frac{\pi_{ii} w_k^{T_i}}{\pi_{ii} w_k^{T_i} + \sum_{\substack{j=1 \\ j\neq i}}^{M} 1 \cdot w_k^{s_{ij}^{\beta ij}}}, \tag{30}$$

and the mixing probability $\mu_k^{j|i}$ for the highest-order static modes is

$$\mu_k^{j|i} = \frac{1 \cdot \bar{w}_k^{s_{ij}^{\beta ij}}}{\pi_{ii} w_k^{T_i} + \sum_{\substack{j=1 \\ j\neq i}}^{M} 1 \cdot w_k^{s_{ij}^{\beta ij}}}. \tag{31}$$

Step II: Mode transfer

The second algorithmic step of the STC-IMM filter is to transfer mode probability mass. As illustrated in Fig. 4, the mode transfer is between static modes of the mode set (with probability 1) and between the transition modes of each model i and the first static mode of the other models $j \neq i$ (with probability π_{ij}).

Step III: Filtering

Step three of the algorithm is the filtering step. The filtering is performed with an ordinary linear or nonlinear model-matched filter, e.g., a Kalman or an Extended Kalman filter. After filtering, the mode probabilities $w_{k+1}^{s_{ij}^l}$ and $w_{k+1}^{T_i}$ are updated according to Bayes' rule (cf. (15)), where the normalization is with respect to all modes of the filter.

Step IV: MMSE estimation

After filtering, the final step is to produce the output of the STC-IMM filter, viz. the MMSE estimate of the target vector $\mathbf{x}_k$. The estimate is given as the weighted sum

$$\hat{\mathbf{x}}_{k|k} = \sum_{i=1}^{M} \left(w_k^{T_i} \hat{\mathbf{x}}_{k|k}^{T_i} + \sum_{\substack{j=1 \\ j \neq i}}^{M} \sum_{l=1}^{\beta^{ij}} w_k^{s_{ij}^l} \hat{\mathbf{x}}_{k|k}^{s_{ij}^l} \right). \tag{32}$$

4.3.2 Comparison with LCMT

Applied to the new model structure, there are some differences between the STC-IMM and the LCMT algorithms. First of all, the STC-IMM algorithm is specifically designed for the proposed model structure, even though it is possible to adapt it to a general pmf, if immediate merging is used (see Section 5.2). Besides that, there are two main differences between the two algorithms, where the first one concerns the spawning of new hypotheses. In the proposed algorithm, there is one hypothesis for each model, i.e., one hypothesis for transition from model 1 to model j, one from model 2 to model j, and so on. For LCMT all these hypotheses are merged into one, which degrades performance. An appealing characteristic of the new filter is that the depth of a model can be different depending from which model the transition occurs. This can not be incorporated in LCMT, due to its merging rule. The LCMT algorithm is hence not as flexible as the presented algorithm.

5 Further simplifications and their implications

To further reduce the computational complexity of the proposed STC-IMM algorithm, approximations are needed. In this section we present two possible approximations that reduce the complexity significantly. Their effect on the state estimation accuracy and further relations to previous sojourn-time dependent algorithms are also discussed.

5.1 Pruning

Experience has shown that during large parts of operation, several of the filter modes have probabilities very close to zero. In the MMSE estimate

of the state vector (see (32)), these filter modes will have a small impact. Hence, by pruning these low-probability branches of the model hypothesis tree, the computational complexity can be reduced, with slight impact on filter performance. The reduction in complexity comes mainly from the fact that model-matched Kalman filters do not have to be run for these modes. In the proposed filter, we introduce a threshold, γ. If a mode probability is below this threshold, we consider it to have no influence on the state vector estimate, and it can thus be pruned. In the filter algorithm, as described in step III of Table 1, pruning is implemented by setting the mode probability to zero, and omitting the model-matched filtering.

The pruning threshold also affects the merging in step I of the algorithm, see Table 1. For a two-model case, if either the transition mode or the highest-order static mode have weight below the threshold, then no mixing is needed, which decreases the computational burden. If the number of models is larger than two, the mixing is instead between a lower number of modes, when some of the mixing modes are pruned.

5.2 Early Merging

The merging introduced in the proposed algorithm was motivated by the fact that the posterior density, after having performed model-matched filtering for β^{ij} time steps, is similar to the posterior density after having performed model-matched filtering for $\beta^{ij}+1$ time steps. To further reduce the computational burden, a natural variation of this approximation is to perform merging at an earlier stage than at the transition mode of each model. However, to preserve the depth of the filter, each mode is still assigned its own probability weight. We let W^{ij} denote how many of the β^{ij} static modes that are to be represented by their own posterior density. The modes of higher order than W^{ij} are all represented by the pdf of the transition mode, but with individual weights. The early merging mixing probabilities are described by (cf. (30)–(31))

$$\tilde{\mu}_k^{T_i} = \frac{\pi_{ii} w_k^{T_i} + \sum_{\substack{j=1 \\ j\neq i}}^{M} \sum_{l=W^{ij}+1}^{\beta^{ij}} w_k^{s_{ij}^l}}{\pi_{ii} w_k^{T_i} + \sum_{\substack{j=1 \\ j\neq i}}^{M} \sum_{l=W^{ij}+1}^{\beta^{ij}} w_k^{s_{ij}^l} + \sum_{\substack{j=1 \\ j\neq i}}^{M} \omega_k^{ij}}, \tag{33}$$

$$\omega_k^{ij} = \begin{cases} w_k^{W^{ij}}, & W^{ij} > 0 \\ \pi_{ji} w_k^{T_j}, & W^{ij} = 0 \end{cases}$$

$$\tilde{\mu}_k^{j|i} = \frac{\sum_{\substack{j=1 \\ j\neq i}}^{M} \omega_k^{ij}}{\pi_{ii} w_k^{T_i} + \sum_{\substack{j=1 \\ j\neq i}}^{M} \sum_{l=W^{ij}+1}^{\beta^{ij}} w_k^{s_{ij}^l} + \sum_{\substack{j=1 \\ j\neq i}}^{M} \omega_k^{ij}}. \tag{34}$$

If the merging time is drawn to its limit, i.e., if $W^{ij} = 0,\ i, j \in \{1, \ldots, M\}$, then only M posterior densities and M model-matched filters are needed, just as for IMM. Still, there are N mode weights (cf. (27)) in total, so the sojourn-time dependence can still be treated, although in a more approximate fashion. Assume that we want to run the SM-IMM filter for the assumed type of sojourn-time pmf. That filter then needs M model-matched filters. Due to the property of the geometrical distribution at the end part of the assumed pmf, the conditional probability of the sojourn times $p\{T_k | m_k, \mathbf{Y}_{1:k}\}$ would only need to consider sojourn times up to $\beta^{ij} + 1$ time steps. Both the proposed and the SM-IMM filters would therefore need to store the same amount of data, and perform similar calculations for merging. However, the SM-IMM algorithm uses the undeclared assumption (cf. [15]) of conditionally independent sojourn times. This assumption is not present in the proposed algorithm, which indicates that SM-IMM is similar, but not equal, to the $W^{ij} = 0,\ i, j \in \{1, \ldots, M\}$ version of the presented algorithm.

6 Selecting design parameters

The proposed system model includes several design parameters such as sojourn time conditions, β^{ij}, the transition probabilities, π_{ij}, and the model-set, f, h. In this section we suggest off-line methods to select suitable values for β^{ij}, π_{ij} and the process model noise parameters. The described methods are used in the filter evaluation. The reader is referred to [20] for an overview on model-set design.

6.1 Optimizing minimum sojourn times and transition probabilities using Monte Carlo simulation

A straightforward approach to find the best β^{ij} is to use Monte Carlo (MC) simulations. To do this, reference or validation data is needed. On the validation data, the filter is run with a set of β^{ij} variables and with a transition probability matrix, $\mathbf{\Pi}$, and the total MSE is calculated. This is then repeated for different sets of β^{ij} and different transition probability matrices, and the best set of parameters are the ones that minimize the MSE. In the search for the best parameters, a simple grid search can be used, provided the number of models is not too large.

6.2 Optimizing motion model parameters based on MCMC samples

For a chosen process model, it is of interest to find the optimal process noise parameters (noise standard deviations). The process noise has the role of describing modeling uncertainties and model errors. Assume that we have some reference data with true states over time, which is to be represented by a motion model. The optimal estimate of the model parameter vector θ, in the MMSE sense, would hence be the MMSE estimate,

$$\hat{\theta}^{\mathrm{MMSE}} = \mathbb{E}\left\{\theta \middle| \mathbf{X}_{1:k}\right\}, \tag{35}$$

where $\mathbf{X}_{1:k} = \{\mathbf{x}_1\ \mathbf{x}_2 \dots \mathbf{x}_k\}$ is the set of all state vectors up to the current time k. With the Markov chain Monte Carlo (MCMC) approach, this estimate is approximately found by sampling the posterior distribution $p(\theta|\mathbf{X}_{1:k})$, using for example the Metropolis-Hastings algorithm [21], which yields samples $\theta^{(m)}$, $m \in \left\{1, 2, \dots, N^{\mathrm{MC}}\right\}$, where N^{MC} is the number of samples. The parameter vector estimate is then given by the Monte Carlo approximation of the MMSE estimate,

$$\hat{\theta}^{\mathrm{MCMC}} = \frac{1}{N^{\mathrm{MC}}} \sum_{m=1}^{N^{\mathrm{MC}}} \theta^{(m)}. \tag{36}$$

Asymptotically, the MMSE and MCMC estimates are identical.

7 Motion models for benchmark scenario

To assess different tracking algorithms and be able to compare them, a set of reference, or benchmark, problems are most valuable. For aircraft tracking, much effort has been put into constructing such scenarios. In the first benchmark scenario [22], a problem for tracking maneuvering targets was presented, involving beam pointing control of a phased array radar. In [17] (with an initial version in [23]), the first benchmark was extended to include the real-world aspects of false alarms and electronic counter-measures. A further extension, commonly referred to as the fourth benchmark, is presented in [24], where the key problems are to track highly maneuvering targets, closely spaced targets and targets in the presence of sea-surface-induced multipath. As the focus of this article is on state estimation, and not on the full tracking problem, we are mainly interested in benchmark trajectories. Since the authors of the second benchmark distribute the computer simulation code

and the trajectory files (in MATLAB format), we use those trajectories[4] for evaluation.

The benchmark scenario considers aircraft tracking. A distinguishing property of aircraft is that they fly straight ahead for long periods of time — a property that is especially pronounced for commercial airliners. A model within the multiple-model framework that captures this type of motion is hence natural. When a target does not perform a maneuver, it is referred to as being in non-maneuver motion. A common model for non-maneuver is the (nearly) constant velocity (CV) model, which is governed by (cf. (3))

$$\mathbf{x}_k^{\mathrm{CV}} = [x \quad y \quad \dot{x} \quad \dot{y}]^T \tag{37}$$

$$\mathbf{A}^{\mathrm{CV}} = \begin{bmatrix} \mathbf{I}_{2\times 2} & T_0\mathbf{I}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \mathbf{I}_{2\times 2} \end{bmatrix} \tag{38}$$

$$\mathbf{B}^{\mathrm{CV}} = \begin{bmatrix} T_0^2/2\mathbf{I}_{2\times 2} \\ T_0\mathbf{I}_{2\times 2} \end{bmatrix} \tag{39}$$

$$\mathbf{Q}^{\mathrm{CV}} = \mathbf{B}^{\mathrm{CV}} \begin{bmatrix} \sigma_{v_x}^2 & 0 \\ 0 & \sigma_{v_y}^2 \end{bmatrix} \left(\mathbf{B}^{\mathrm{CV}}\right)^T \tag{40}$$

where $\sigma_{v_x}^2$ and $\sigma_{v_y}^2$ are the acceleration noise variances in the x- and y directions, respectively, $\mathbf{Q}^{\mathrm{CV}}$ is the covariance matrix of the noise vector $\mathbf{v}^{\mathrm{CV}}$, and T_0 is the measurement interval.

A second distinguishing part of an aircraft's dynamics is its turns. While the motion in non-maneuver is uncorrelated between tracking directions, the motion during a coordinated turn is highly correlated [19]. For such types of maneuvers, several coordinated turn models have been developed. In benchmark scenario studies, coordinated turn models for horizontal turns are common. These models are mainly of two types, where the difference between them lies in the state vector. The common element of the state vectors is the turn rate parameter, ω, which is introduced in order to neatly treat the (nearly) constant turn rate of a coordinated turn. A definition of turn rate is found in [19]. Often the 3D turn rate is replaced with the simplified horizontal-plane turn rate, obtained by letting the vertical velocity and acceleration be zero. For horizontal turn models with velocity states, the state vector is the same as for the CV model, but with ω as extra state. The corresponding non-linear motion model is found in [19, 25].

For a coordinated turn, it is often assumed that the speed is nearly constant. By letting speed be a part of the state vector, this nearly constant speed assumption can be handled, since perturbations in speed can be inflicted through a dedicated noise component for speed. Following the pro-

[4]These trajectories are almost identical to those of the first benchmark.

posal in [26], a nearly constant speed horizontal turn model is derived for the state vector (HTS stands for nearly constant speed horizontal turn model)

$$\mathbf{x}^{\mathrm{HTS}} = \begin{bmatrix} x & y & v & \varphi & \omega \end{bmatrix}^T , \tag{41}$$

where φ is heading angle, defined in a target-oriented coordinate system. Described as

$$\mathbf{x}^{\mathrm{HTS}}[k+1] = \mathbf{f}^{\mathrm{HTS}}\left(\mathbf{x}^{\mathrm{HTS}}[k]\right) + \mathbf{v}^{\mathrm{HTS}}[k+1], \tag{42}$$

where $\mathbf{v}^{\mathrm{HTS}}[k+1] \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}^{\mathrm{HTS}})$, the HTS motion model is governed by

$$\mathbf{f}^{\mathrm{HTS}}\left(\mathbf{x}^{\mathrm{HTS}}\right) = \begin{bmatrix} x + \frac{2v}{\omega}\sin(\frac{\omega T_0}{2})\cos(\varphi + \frac{\omega T_0}{2}) \\ y + \frac{2v}{\omega}\sin(\frac{\omega T_0}{2})\sin(\varphi + \frac{\omega T_0}{2}) \\ v \\ \varphi + \omega T_0 \\ \omega \end{bmatrix} \tag{43}$$

$$\mathbf{Q}^{\mathrm{HT}} = \begin{bmatrix} \mathbf{0}_{2\times1} & \mathbf{0}_{2\times1} & \mathbf{0}_{2\times1} & \mathbf{0}_{2\times1} & \mathbf{0}_{2\times1} \\ 0 & 0 & T_0\sigma_v^2 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{3}T_0^3\sigma_\omega^2 & \frac{1}{2}T_0^2\sigma_\omega^2 \\ 0 & 0 & 0 & \frac{1}{2}T_0^2\sigma_\omega^2 & T_0\sigma_\omega^2 \end{bmatrix}. \tag{44}$$

For military aircraft with high agility, the start and end of turns are not smooth as for commercial airliners, but are instead often more abrupt. When a turn is initiated, the acceleration in latitudinal (cross-velocity) direction is rapidly increased. For such maneuvers it is advantageous to have a separate model, that aids the turn model in start and termination of turns. Since acceleration changes are the core of this maneuver, the model is well suited to have acceleration as a state variable, with an associated jerk process noise that describes its perturbations. Hence, a third-order process model in target-centered coordinates for start/termination of turns would be a useful tool for agile maneuvers. We have developed a new start/termination model with these properties. The following state vector is used for the model (CAL stands for constant acceleration model in local coordinate system)

$$\mathbf{x}_k^{\mathrm{CAL}} = \begin{bmatrix} x & y & v & \varphi & \ddot{x}^l & \ddot{y}^l \end{bmatrix}^T . \tag{45}$$

The states $\ddot{x}^l$ and $\ddot{y}^l$ represent acceleration in the target velocity and cross-velocity directions, respectively. Using the discretized linearization approach[5]

[5]This approach is also used to find the matrix for the covariance update step of the EKF filter.

[27], the non-linear process model describing the constant-acceleration movement in this coordinate system is

$$\begin{aligned}\mathbf{x}^{\mathrm{CAL}}[k+1] \cong \mathbf{x}^{\mathrm{CAL}}[k] + \mathbf{G}\mathbf{f}^{\mathrm{CAL}}\left(\mathbf{x}^{\mathrm{CAL}}[k]\right) \\ + \mathbf{G}\mathbf{B}\mathbf{v}[k+1]\end{aligned} \tag{46}$$

$$\mathbf{G} = \left(\sum_{i=0}^{q-1} \left(\nabla \mathbf{f}^{\mathrm{CAL}}\left(\mathbf{x}^{\mathrm{CAL}}[k]\right)^T \right)^i \frac{T_0^{i+1}}{(i+1)!} \right) \tag{47}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{0}_{4\times 2} \\ \mathbf{I}_{2\times 2} \end{bmatrix} \tag{48}$$

where $\mathbf{G}$ is a 6×6 matrix, $\mathbf{v}[k+1] \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}^{\mathrm{CAL}})$, and where

$$\nabla \mathbf{f}^{\mathrm{CAL}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_2}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_1}{\partial x_n} & \frac{\partial f_2}{\partial x_n} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \tag{49}$$

is the gradient of the (n-dimensional) non-linear function $\mathbf{f}^{\mathrm{CAL}}$, which is the continuous-time process model function, given by

$$\mathbf{f}^{\mathrm{CAL}}\left(\mathbf{x}^{\mathrm{CAL}}\right) = \begin{bmatrix} v\cos(\varphi) & v\sin(\varphi) & \ddot{x}^l & \dddot{y}^l/v & 0 & 0 \end{bmatrix}^T. \tag{50}$$

Finally, the covariance matrix is

$$\mathbf{Q}^{\mathrm{CAL}} = \tilde{\mathbf{G}}\mathbf{Q}\tilde{\mathbf{G}}^T, \tag{51}$$

where

$$\tilde{\mathbf{G}} = \mathbf{G}\mathbf{B} = \begin{bmatrix} T_0^3/6\cos(\varphi) & -T_0^3/6\sin(\varphi) \\ T_0^3/6\sin(\varphi) & T_0^3/6\cos(\varphi) \\ T_0^2/2 & 0 \\ -\left(T_0^3\,\dddot{y}^l\right)/\left(6v^2\right) & T_0^2/\left(2v\right) \\ T_0 & 0 \\ 0 & T_0 \end{bmatrix} \tag{52}$$

$$\mathbf{Q} = \begin{bmatrix} \sigma_{\ddot{x}^l}^2 & 0 \\ 0 & \sigma_{\dddot{y}^l}^2 \end{bmatrix}. \tag{53}$$

The elements $\sigma_{\ddot{x}^l}^2$ and $\sigma_{\dddot{y}^l}^2$ of the covariance matrix are the along-velocity and cross-velocity jerk noise variances, respectively. Since the jerk noise effect on the position states (x and y) depend on the heading angle φ, the covariance matrix becomes geometry dependent (through $\tilde{\mathbf{G}}$), in contrast to the previously described models.

7.1 Model selection

For the three benchmark scenarios, there have been several solutions, most of which have included an IMM filter with a set of motion models. A solution to the first benchmark is given in [28]. To the second benchmark, IMM-based solutions are presented in [29, 30] and [31] (with an initial version in [32]). In [33] (with an initial version in [34]) an IMM/MHT solution is given, and in [35] an IMM-based solution to the fourth benchmark is presented. All the solutions to the first two benchmark scenarios use three models for the IMM filter, with the same properties, viz. one model for non-maneuver, one model for turns (maneuver) and one model for start and end of turns. For non-maneuver, a standard CV model with small process noise is used. For maneuver, the solutions differ in that they either use a turn model (3D turn model in [29] and 2D horizontal turn models in [30], [33]) or a CV model with a large process noise ([28] and [31]). For the start/end of turns [28], [31] and [33] use a CA model with large process noise, [29] a CA model with acceleration parallel to velocity only and [30] a general Singer model.

With the same argumentation as in the above papers, we have chosen a CV model for non-maneuver. For maneuver (turns), we use the previously described CT model (HTS). Finally, for start/end of turns we use the new CA model in local coordinates with acceleration parallel to the velocity vector. All filters operate in a horizontal plane, which means that we assume that altitude is tracked separately, just as in those benchmark solutions where horizontal turn models are used. One may direct some criticism towards treating altitude separately, but actually [30] shows that a horizontal turn model combined with an altitude tracker performed better overall than a full 3D turn model for the second benchmark.

The covariance and state mixing require that the state vectors and covariance matrices of the different models are first transformed to the state space of the model. Hence, six different transformation matrices are needed for the covariance matrices and six transformation functions are needed for the state vectors. Furthermore, the presentation requires a common coordinate system, which in our case is the 4-state Cartesian coordinates (x, y, $\dot{x}$, $\dot{y}$) (the CV model coordinates). The transformation matrices are calculated as

$$\mathbf{F}^{\mathrm{C1}\to\mathrm{C2}} = \nabla_{\mathbf{x}^{C1}} \left(\mathbf{x}^{C2} \right), \tag{54}$$

for transformation from model C1 to model C2, where C1 and C2 are any of the CV, CAL or CT models, and where the gradient is with respect to $\mathbf{x}^{C1}$.

Table 2: Optimal transition probability matrices for STC-IMM and IMM. Models 1 to 3 are the CV, CT and CAL models, respectively.

STC-IMM			IMM		
0.965	0.03	0.005	0.96	0.02	0.02
0.09	0.85	0.06	0.01	0.98	0.01
0.08	0.4	0.52	0.15	0.45	0.4

Table 3: Optimal depths of the STC-IMM filter.

β^{21}	β^{31}	β^{12}	β^{32}	β^{13}	β^{23}
6	6	8	8	0	0

8 Results

8.1 Parameter optimization results

The parameters β^{ij} and the transition probability matrix $\mathbf{\Pi}$ were optimized jointly over the six benchmark trajectories, such that the best overall parameter set was found. This approach was chosen since optimization to each and every trajectory is not a feasible approach in practise. To find the optimal parameters, a line search method was used where only some of the parameters were varied in each run to limit the computational complexity. For each optimization step, 10 Monte Carlo runs were used. For the STC-IMM algorithm, the number of variables to optimize was 9, while it was 6 for IMM. To further reduce the complexity, the depth of each model was set equal regardless of from which model transition occurs. That is, $\beta^{12} = \beta^{13}$ for model 1, and equivalently for the two other models. Finally, the cost function of the optimization was the mean-squared error (MSE).

The results are summarized in Tables 2 and 3. Model 1 to 3 corresponds to the CV, CT and CAL models, respectively. As we can see, the depths of the proposed filter for the start/end of turns model is zero, which is what we expect, since it only has the purpose of kick-starting or ending turns. Since some of the trajectories have short non-maneuver segments, the depth of non-maneuver is shorter than for maneuver, which was not expected prior to the optimization.

For the process noise parameters of the CV and CT models, an MCMC approach was taken to approximately evaluate the MMSE estimate of the parameter vector, jointly over the six trajectories. The joint likelihood func-

Table 4: Optimal process noise parameters.

σ_{v_x}	σ_{v_y}	$\sigma_{\dot{v}}$	$\sigma_{\dot{\omega}}$	$\sigma_{\ddot{x}^l}$	$\sigma_{\ddot{y}^l}$
1.34	1.21	5.61	0.021	0	29.3

tion of each model is the multiplication of all one-step prediction functions of the same model. As prior, a non-informative Jeffrey's prior was used. For the start/end of turns model, the jerk standard deviation $\sigma^2_{\ddot{y}^l}$ in the cross-velocity direction was estimated as the mean value over the 32 cross-velocity jerks that occur for the 6 trajectories. The along-velocity jerk was set to 0. Table 4 summarizes the optimization results for the six parameters. The difference in σ_{v_x} and σ_{v_y} has to do with the geometry of the trajectories.

8.2 Filtering performance on benchmark problems

The six benchmark trajectories regarding X and Y position are shown in Fig. 5. Note that some short-period agile maneuvers are not seen in the figures. The first target is a large aircraft (like a military cargo aircraft) and the second one represents a smaller, more maneuverable one (like a Learjet). Target three and four represent medium bombers with high speed and good maneuverability. Finally, target five and six represent fighter/attack aircraft with high speed and agility.

In the evaluation of filter performance for the proposed filter and IMM, the jointly optimal parameters of section 8.1 were used. The performance measure was the root mean-squared error (RMSE) in position, after measurement update. However, also the one-step prediction error was calculated (called pRMSE), since it is often more interesting to know the size of this error, due to its close connection to data association. In the evaluation of the filters, 1000 Monte Carlo runs were used for each trajectory and the total performance measure is the mean value of the average position error. Measurements were presented to the filter as position in X and Y coordinates, where the measurement covariance matrix $\mathbf{R}$ is diagonal with σ_r^2 as non-zero elements, for $\sigma_r = 100$m.

In Tables 5 and 6 the RMSE after measurement update and for prediction, respectively, are presented for the two filters, as well as the relative performance increase with the proposed filter. As we can see, the proposed filter performs better than IMM for all trajectories. The overall performance gain is 9.3% for prediction and 7.4% after measurement update. The largest gain is for the high performance commercial aircraft, which is not surprising

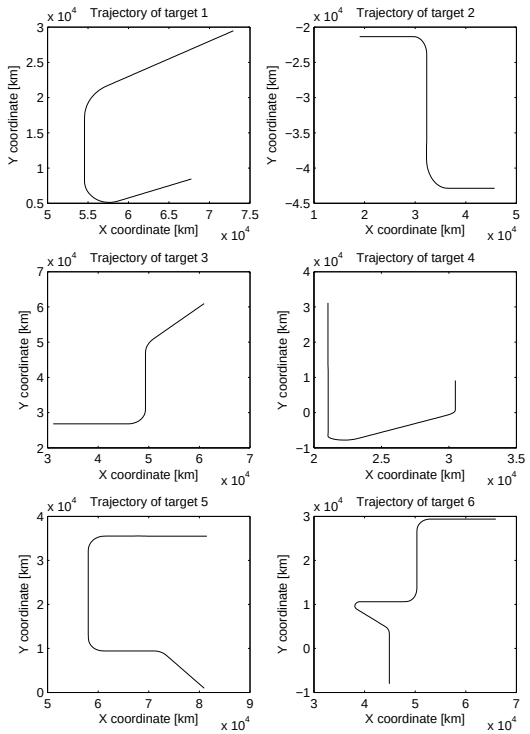

Figure 5: The six benchmark trajectories.

Table 5: Mean RMSE for proposed filter and IMM after measurement update, for the six trajectories and the average over all trajectories.

	New filter mean RMSE [m]	IMM mean RMSE [m]	Gain new contra IMM (MSE)
Traj. 1	70.01	73.44	9.13%
Traj. 2	71.94	77.00	12.7%
Traj. 3	86.90	89.37	5.45%
Traj. 4	82.13	85.55	7.85%
Traj. 5	103.0	104.6	2.87%
Traj. 6	100.3	106.4	9.42%
Overall	86.65	90.06	7.43%

Table 6: Mean prediction RMSE for proposed filter and IMM, for the six trajectories and the average over all trajectories.

	New filter mean pRMSE [m]	IMM mean pRMSE [m]	Gain new contra IMM (MSE)
Traj. 1	90.43	94.87	9.14%
Traj. 2	97.48	103.8	11.7%
Traj. 3	116.5	120.7	6.88%
Traj. 4	109.8	116.8	11.6%
Traj. 5	145.8	151.7	7.65%
Traj. 6	147.6	155.9	10.4%
Overall	119.9	126.6	9.4%

since it is in non-maneuver and maneuver for long periods of time — agreeing well with the model assumption. The same holds for the large aircraft (trajectory 1). A bit unexpected, however, is the fact that the performance gain is close to 10% for the second fighter/attack aircraft, which implies that even agile targets agree well with the assumptions of this article, if the maneuvers are not too short in time. The performance increase is lower for trajectory 5, since the target it resembles performs several quick and short-lasting maneuvers, which does not agree well with the proposed filter assumptions.

In Figures 6-7, the mean position errors after measurement update as a function of time are presented for the six trajectories — both for IMM and the proposed filter. We see from the figures that the performance gain comes mainly from the non-maneuver parts of the trajectories (which is seen in the figures as the parts of low RMSE), which is in agreement with the study in [1]. The filter performance improvement with the new approach is

hence larger when the non-maneuver segments are longer. As we can see from Fig. 6, the filter performance gain with the new approach for trajectory 2 is the largest, since the second non-maneuver segment lasts for so long; compare for example with the second non-maneuver segment of trajectory 1 in Fig. 6. In the maneuver sequences, the IMM performs better on some segments, which is explained by the fact that the parameter optimization has the overall MSE as objective function. Since the performance gain comes mainly from non-maneuver segments, the best parameters are the ones which maximize performance in non-maneuver, while still allowing for good tracking in maneuver.

Apart from filter performance, it is also of interest to study the average filter model probabilities of the trajectories over time. We study them only for trajectories 2 and 5, to see if there is any difference, for example in the probability of the start/end of maneuver model. In Figures 8-9, these two average model probability sequences are given for IMM and STC-IMM. From the figures we see a difference between the two filters. While the probability of the CA model (start/end of turns) is at a constant low level for IMM in trajectory two, it is very close to zero at non-maneuver for the proposed filter, while the probability increases in the beginning of a maneuver. So the model works more as expected together with the proposed filter. The reason for the probability not going to zero immediately is that the transition probability from the maneuver to the CA model is 0.06, which is quite large. But one sees that the probability decreases after the initiation of a turn, and then again increases at the termination of the turn. For the agile target (Fig. 9), the same tendencies are seen, even though the CA model is active for two of the turn initiations of the IMM filter. It thus seems as if the IMM filter does not take advantage of the start/end of turns model in the same way as does the STC-IMM filter.

Another conclusion that is drawn by observing the model probabilities is that the proposed filter is faster in detecting the transition from maneuver to non-maneuver. It is seen in the figures as a more rapid increase in non-maneuver model probability after a maneuver.

9 Conclusions

In this article we present a new multiple model filter which utilizes the property that target dynamics often persist for some time. By taking this inertial property into account, the performance compared to a conventional multiple-model filter is improved. The assumption made in the article is that model sojourn times have a mass function that is zero up to a model-specific time,

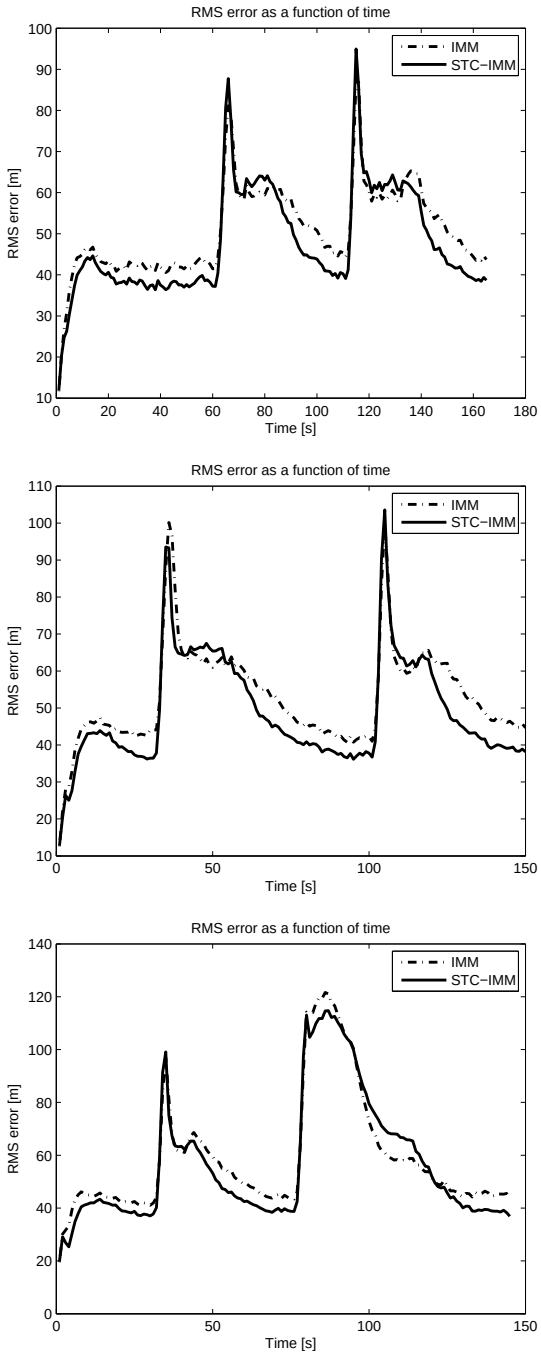

Figure 6: Mean RMS error as a function of time for trajectory 1 [top], trajectory 2 [middle], and trajectory 3 [bottom].

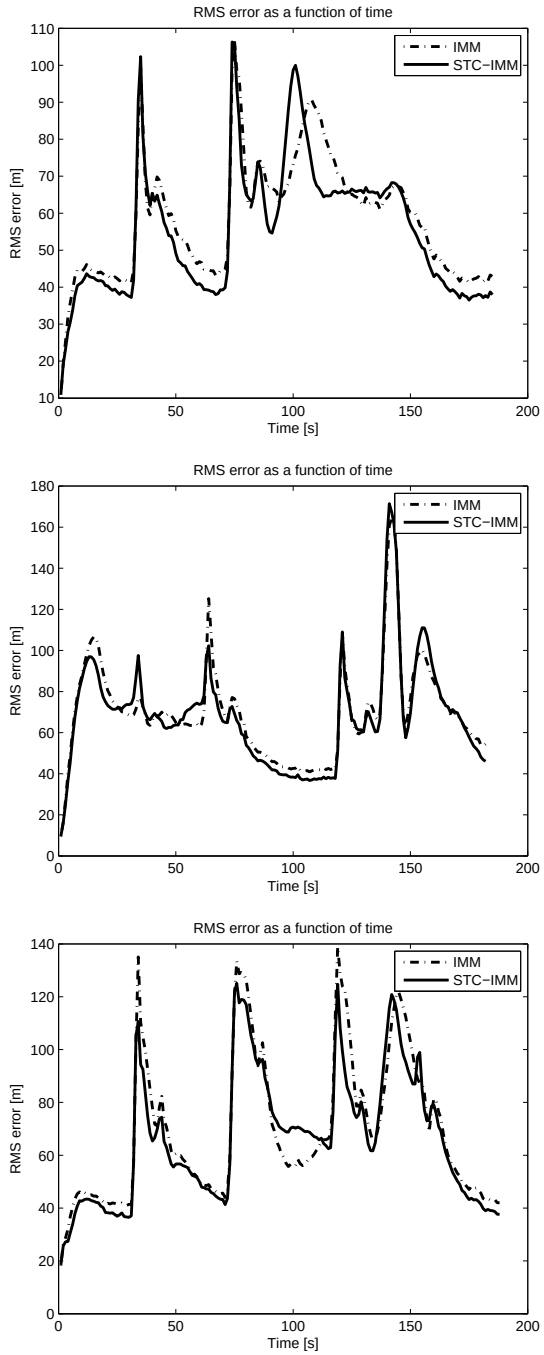

Figure 7: Mean RMS error as a function of time for trajectory 4 [top], trajectory 5 [middle], and trajectory 6 [bottom].

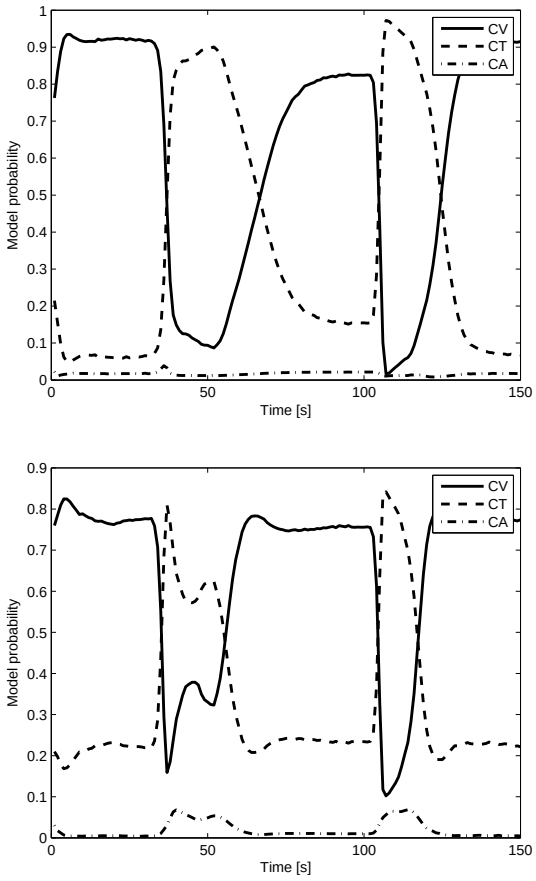

Figure 8: Average model probability for IMM [top] and STC-IMM [bottom] as a function of time for trajectory 2 (high performance commercial aircraft).

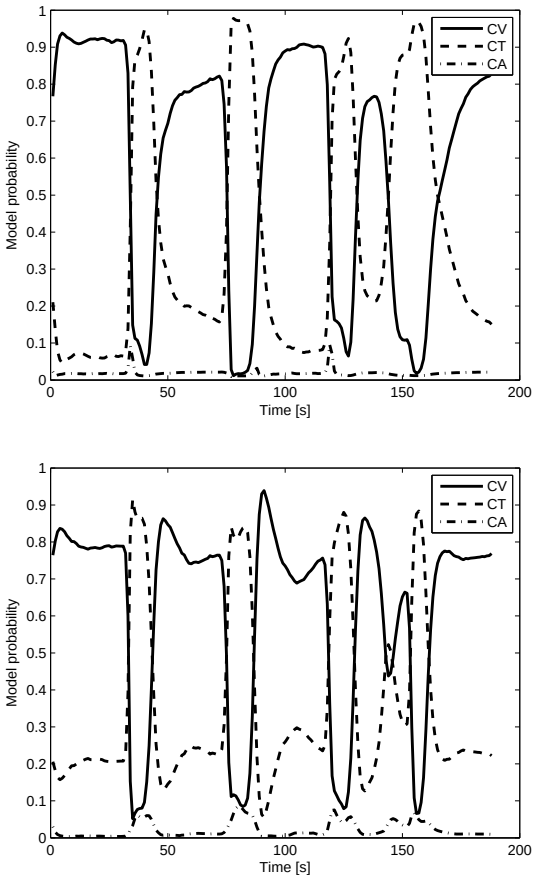

Figure 9: Average model probability for IMM [top] and STC-IMM [bottom] as a function of time for trajectory 6 (fighter/attack aircraft).

and then follows a geometrical distribution. The mass function is thus described by only two parameters.

The proposed filter is evaluated on six benchmark scenarios for tracking. The results show that the proposed filter performs better than IMM for all six trajectories of the benchmark. The trajectories do not agree with either the assumptions of the article or the IMM assumptions. However, there are inertial properties in the trajectories, which means that dynamics persist for some time before changes occur. This is utilized by the proposed filter. When the target is in non-maneuver mode, the maneuver model does not affect the non-maneuver model, because of the filter depth. Hence, the high-probability non-maneuver pdf is more accurately described by the proposed algorithm than by IMM. This improved description enhances performance.

References

[1] L. Svensson and D. Svensson, "Multiple model filtering with switch time conditions," in *Proceedings of the 10th International Conference on Information Fusion*, July 2007.

[2] H. Blom, "A sophisticated tracking algorithm for atc surveillance radar data," in *Proceedings of the International Conference on Radar*, May 1984, pp. 393–398.

[3] ——, "An efficient filter for abruptly changing systems," in *Proceedings of the 23rd IEEE Conference on Decision and Control*, vol. 23, December 1984, pp. 656–658.

[4] H. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with markovian switching coefficients," *IEEE Transactions on Automatic Control*, vol. 33, no. 3, pp. 780–783, July 1988.

[5] G. Ackerson and K. Fu, "On state estimation in switching environments," *IEEE Transactions on Automatic Control*, vol. AC-15, no. 1, pp. 10–17, February 1970.

[6] C. Chang and M. Athans, "State estimation for discrete systems with switching parameters," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-14, no. 3, pp. 418–424, May 1978.

[7] J. Tugnait, "Detection and estimation for abruptly changing systems," *Automatica*, vol. 18, no. 5, pp. 607–615, September 1982.

[8] F. Gustafsson, *Adaptive Filtering and Change Detection.* Chichester, UK: Wiley, 2000.

[9] X. Li and V. P. Jilkov, "Survey of maneuvering target tracking. part i: Dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, October 2003.

[10] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems.* Boston, USA: Kluwer Academic, 1999.

[11] P. Anselone, "Ergodic theory for discrete semi-markov chains," *Duke Math. J.*, vol. 27, no. 1, pp. 33–40, 1960.

[12] R. Howard, *Dynamic Probabilistic Systems.* New York: Wiley, 1971, vol. 2.

[13] L. Campo, P. Mookerjee, and Y. Bar-Shalom, "Failure detection via recursive estimation for a class of semi-markov switching systems," in *Proceedings of the 27rd IEEE Conference on Decision and Control*, December 1988.

[14] ——, "State estimation for systems with sojourn-time-dependent markov model switching," *IEEE Transactions on Automatic Control*, vol. 36, no. 2, pp. 238–243, February 1991.

[15] H. Blom, "Hybrid state estimation for systems with semi-markov switching coefficients," in *Proceedings of the European Control Conference*, July 1991.

[16] A. Zubov and A. Petrov, "Estimation in nonlinear stochastic systems with sudden changes in the structure and state coordinates," *Soviet Journal of Computer and Systems Sciences (Tekhnicheskaya Kibernetika)*, vol. 29, no. 3, pp. 9–21, May-June 1991.

[17] W. Blair, G. Watson, T. Kirubarajan, and Y. Bar-Shalom, "Benchmark for radar allocation and tracking in ecm," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 4, pp. 1097–1114, October 1998.

[18] A. Petrov and A. Zubov, "On applicability of the interacting multiple-model approach to state estimation for systems with sojourn-time-dependent markov model switching," *IEEE Transactions on Automatic Control*, vol. 41, no. 1, pp. 136–140, January 1996.

[19] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood, MA: Artech House, 1999.

[20] X. Li and V. Jilkov, "Survey of maneuvering target tracking. part v: Multiple-model methods," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1255–1321, October 2005.

[21] W. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, April 1970.

[22] W. Blair, G. Watson, and S. Hoffman, "Benchmark problem for beam pointing control of phased array radar against maneuvering targets," in *Proceedings of the American Control Conference*, June 1994.

[23] W. Blair, G. Watson, G. Gentry, and S. Hoffman, "Benchmark problem for beam pointing control of phased array radar against maneuvering targets in the presence of ecm and false alarms," in *Proceedings of the American Control Conference*, June 1995.

[24] G. Watson and D. McCabe, "Benchmark problem with a multisensor framework for radar resource allocation and tracking of highly maneuvering targets, closely-spaced targets, and targets in the presence of sea-surface-induced multipath," NSWC Dahlgren, VA, USA, Tech. Rep. NSWC-DD/TR-99/32, March 1999.

[25] G. Watson and W. Blair, "IMM algorithm for tracking targets that maneuver through coordinated turns," in *Proceedings of the SPIE – The International Society for Optical Engineering*, vol. 1698, April 1992, pp. 236–247.

[26] J. Gertz, "Multisensor surveillance for improved aircraft tracking," *Lincoln Laboratory Journal*, vol. 2, no. 3, pp. 381–396, 1989.

[27] F. Gustafsson and A. J. Isaksson, "Best Choice of Coordinate System for Tracking Coordinated Turns," in *Proceedings of the 35^{th} Conference on Decision and Control*, December 1996.

[28] E. Daeipour, Y. Bar-Shalom, and X. Li, "Adaptive beam pointing control of a phased array radar using an imm estimator," in *Proceedings of the American Control Conference*, June 1994.

[29] G. Watson and W. Blair, "Solution to second benchmark problem for tracking maneuvering targets in the presence of false alarms and ecm," in *Proceedings of the SPIE – The International Society for Optical Engineering*, vol. 2561, September 1995.

[30] M. Busch and S. Blackman, "Evaluation of IMM filtering for an air defense system application," in *Proceedings of the SPIE – The International Society for Optical Engineering*, vol. 2561, July 1995, pp. 435–447.

[31] T. Kirubarajan, Y. Bar-Shalom, W. Blair, and G. Watson, "Immpdaf for radar management and tracking benchmark with ecm," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 4, pp. 1115–1133, October 1998.

[32] T. Kirubarajan, Y. Bar-Shalom, and E. Daeipour, "Adaptive beam pointing control of a phased array radar in the presence of ecm and false alarms using immpdaf," in *Proceedings of the American Control Conference*, June 1995.

[33] S. Blackman, R. Dempster, M. Busch, and R. Popoli, "Imm/mht solution to radar benchmark tracking problem," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 2, pp. 730–738, April 1999.

[34] R. Popoli, S. Blackman, and M. Busch, "Application of multiple hypothesis tracking to agile beam radar tracking," in *Proceedings of the SPIE – The International Society for Optical Engineering*, vol. 2759, May 1996.

[35] A. Sinha, T. Kirubarajan, and Y. Bar-Shalom, "Tracker and signal processing for the benchmark problem with unresolved targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 279–300, January 2006.

Paper II

Set JPDA Filter for Multi-Target Tracking

L. Svensson, D. Svensson, M. Guerriero and P. Willett

Submitted to
IEEE Transactions on Signal Processing.

Preliminary results have been published as:

L. Svensson, D. Svensson and P. Willett,
"Set JPDA filter for tracking unordered sets of targets",
in *Proceedings of the 12th International Conference on Information Fusion*, Seattle, USA, 2009.

Set JPDA Filter for Multi-Target Tracking

Lennart Svensson, Daniel Svensson*, Marco Guerriero†
and Peter Willett‡, *Fellow, IEEE*

Abstract

In this article we show that when targets are closely spaced, traditional tracking algorithms can be adjusted to perform better under a performance measure that disregards identity. More specifically, we propose an adjusted version of the Joint Probabilistic Data Association (JPDA) filter, which we call the Set JPDA (SJPDA) filter. Through examples and theory we motivate the new approach, and show its possibilities. To decrease the computational requirements, we further show that the SJPDA filter can be formulated as a continuous optimization problem which is fairly easy to handle. Optimal approximations are also discussed, and an algorithm, KLSJPDA, which provides optimal Gaussian approximations in the Kullback-Leibler sense is derived. Finally, we evaluate the SJPDA filter on two scenarios with closely spaced targets, and compare the performance in terms of the mean Optimal Subpattern Assignment (MOSPA) measure with the JPDA filter, and also with the Gaussian-mixture CPHD filter. The results show that the SJPDA filter performs substantially better than the JPDA filter, and almost as well as the more complex GM-CPHD filter.

Index Terms

Target tracking, filtering theory, random finite set theory, recursive estimation, Bayes methods.

*Lennart Svensson and Daniel Svensson are with the Department of Signals and Systems, Chalmers University of Technology, SE-412 96, Gothenburg, Sweden (E-mail: {lennart.svensson,daniel.svensson}@chalmers.se).

†Marco Guerriero is with Research and Advanced System Design, Elt. Elettronica, S.p.A., 00 131 Rome, Italy (E-mail: marco.guerriero@elt.it).

‡Peter Willett is with the Electrical and Computer Engineering Department, University of Connecticut, Storrs, CT 06269-2157, USA (E-mail: willett@engr.uconn.edu).

1 Introduction

Traditional target tracking algorithms are designed to track targets over time, and discriminate between them by labeling them at each time instant. Examples of such algorithms are the Probabilistic Data Association (PDA) and Joint PDA (JPDA) filters [1,2], and the Multiple Hypothesis Tracking (MHT) algorithm [3–6]. In many applications, such as those in which evidence of target type is being accumulated, the identity (labeling) of the targets is of great importance; but in other cases where one might say that "a threat is a threat", it is not. In this article we show that by not considering target identity, the traditional algorithms can be significantly improved both in terms of density approximations and of estimation accuracy, when evaluated with a metric that disregards target identity.

For the traditional algorithms, the aim is to minimize the Mean Square Error (MSE) between target states and corresponding track estimates. When target identity is not of interest, minimization under such a measure subjects itself to an unnecessary constraint. Instead we need a measure which only describes how good an estimate is in determining where the targets are, regardless of which is which. In this article, we propose the use of the Optimal Subpattern Assignment (OSPA) metric [7]. More specifically, we study the Mean OSPA (MOSPA). The tracking problem can thus be formulated as the minimization of the MOSPA which, as we will see, is significantly different from the problem of minimizing the MSE.

A framework suitable for the description of tracking without target identity is Finite Set Statistics (FISST) [8]. Within that framework, the aim is to track an unordered, or unlabeled, set of targets, described as a Random Finite Set (RFS). Two popular algorithms have been derived within FISST, namely the Probability Hypothesis Density (PHD) and the Cardinalized PHD (CPHD) filters [9–12]. We notice two drawbacks with FISST-based techniques. First, it is difficult to find an analytical expression for the posterior density of the RFS. Therefore, the standard technique of calculating and approximating the posterior density is not applicable. Instead, the filters operate on a first-order approximation of the density of unordered targets, referred to as the intensity function. Second, the first-order approximation has the effect that all targets be assumed independent and identically distributed. Due to these drawbacks, we believe that it is relevant to search for alternative approaches. In the current article, one such alternative is suggested, where the advantages of FISST methods and those of classic recursive filtering are combined, and where it is shown how traditional algorithms can be improved for the problem of minimizing MOSPA.

The approach that we propose relies on the fact that there is a relation

between an ordered density and the density of a random finite set, here called the unordered density, or set density. In fact, there is an infinite number of ordered densities that correspond to the same set density. We refer to a set of such ordered densities as an RFS family. Since optimal MOSPA estimates can be derived from the set density, any ordered density in the corresponding RFS family can be used without losing optimality. We thus have a previously unrecognized possibility of switching ordered densities within traditional tracking filters, to obtain better approximations and performance, when target identity is irrelevant. This is useful since the different densities in the RFS family are more or less suitable to use in practice.

Most tracking algorithms rely on Gaussian approximations. That is, the posterior density is often described as a Gaussian mixture, where the number of mixture components is controlled. In the JPDA filter, which is the filter that we specifically consider in this article, the posterior density is approximated by a single Gaussian in each iteration. By combining traditional methods and FISST techniques, we show that this Gaussian approximation can be made more accurate, when target identity is not of interest. The approach is to utilize the possibility of switching from the original Gaussian mixture, to another density in the same RFS family that can be better approximated as Gaussian. We further show that there is an optimal way, in Kullback-Leibler sense, of making that density switch. From this, we develop a refined JPDA algorithm for optimal approximations, which we call Kullback-Leibler Set JPDA (KLSJPDA). The drawback with the algorithm is its computational demand. We thus also develop a more computationally efficient algorithm, called the Set JPDA (SJPDA). The SJPDA algorithm, which is also based on density switches, optimizes MOSPA performance under the constraint of remaining within the family of Gaussian mixtures, and this generally also leads to a density which is better approximated as Gaussian. Through examples and theory we discuss the reason for that. The details about the optimization criterion for the SJPDA algorithm, and its relation to optimal MOSPA estimates, are found in [13]. Throughout the article, we make the assumption that the number of targets is known.

The article is outlined as follows. In Section II, the problem formulation is stated. Section III concerns two conceptual solutions to the formulated problem, based on traditional approaches and FISST, respectively. In Section IV, a conceptual solution based on the new approach is presented. We also discuss how and why traditional tracking methods can be improved when MOSPA is the target cost function. Section V considers optimal Gaussian approximations in the Kullback-Leibler sense, and the derivation of the KLSJPDA algorithm. In Section VI, the SJPDA filter is derived, and the optimization step is formulated as a continuous optimization problem. Section

VII regards evaluations of the proposed KLSJPDA and SJPDA algorithms. For the SJDPA algorithm, we see that the tracking performance in terms of MOSPA is better than that of the JPDA filter, and almost as good as that of the more complex Gaussian mixture CPHD filter [12]. We also see that the track loss probability is dramatically decreased, which leads to a much longer track life. For the KLSJPDA algorithm we observe similar performance as the SJPDA filter, but the computational complexity is much larger. In Section VIII the article is concluded.

2 Problem formulation

2.1 Target modeling assumptions

In this article we study the problem of tracking an arbitrary, but known and fixed, number of targets. Further, we are not interested in the identities of the targets. To reflect this, we are using a different cost function from the standard formulation, in which MSE is used.

To formulate the problem, we introduce the vector of ordered (labeled) target states

$$\mathbf{X}_k = \left[\left(\mathbf{x}_k^{(1)}\right)^T \left(\mathbf{x}_k^{(2)}\right)^T \ \dots \ \left(\mathbf{x}_k^{(n)}\right)^T \right]^T, \tag{1}$$

where $\mathbf{x}_k^{(i)}$ is the state vector of target number i at time k, and where n is the number of targets in the scene. We further introduce the collection of measurements $\mathbf{Z}^k$,

$$\mathbf{Z}^k = \left\{ \mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_k \right\}, \tag{2}$$

up to the current time step k, where $\mathbf{Z}_k$ is a matrix of measurement vectors at time k.

For the general case, the process model is governed by

$$\mathbf{x}_k = f_{k-1}\left(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}\right), \tag{3}$$

where f_{k-1} is a nonlinear (prediction) function and $\mathbf{v}_{k-1}$ is a realization of the process noise, which has some assumed distribution. As seen, we sometimes omit the superscript on the state vectors, and write $\mathbf{x}_k$ instead of $\mathbf{x}_k^{(i)}$. The corresponding linear-Gaussian motion model is written as

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{v}_{k-1}, \tag{4}$$

where $\mathbf{v}_{k-1} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{Q}_{k-1}\right)$.

The general measurement model is given by

$$\mathbf{z}_k = h_k\left(\mathbf{x}_k, \mathbf{w}_k\right) \tag{5}$$

and the corresponding linear-Gaussian model by

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{w}_k, \tag{6}$$

where $\mathbf{w}_k \sim \mathcal{N}\big(\mathbf{0}, \mathbf{R}_k\big)$ is measurement noise. The measurement set $\mathbf{Z}_k$ does not only include target-generated measurements, but also spurious measurements due to false alarms and clutter. If we let m_k be the number of target-generated measurements (smaller than, or equal to, n), and c_k be the number of clutter measurements at time k, the measurement set is given by

$$\mathbf{Z}_k = \big\{\mathbf{z}_k^{(1)}, \ldots, \mathbf{z}_k^{(m_k)}, \mathbf{z}_k^{c,(1)}, \ldots, \mathbf{z}_k^{c,(c_k)}\big\}. \tag{7}$$

The target-generated measurements, $\mathbf{z}_k^{(1)}, \ldots, \mathbf{z}_k^{(m_k)}$, are governed by (5) or (6), and the clutter measurements, $\mathbf{z}_k^{c,(1)}, \ldots, \mathbf{z}_k^{c,(c_k)}$, follow a density function which is target-state independent.

2.2 MOSPA measure and optimal estimates

The ubiquitous measure in the literature is the squared error (SE)

$$\begin{aligned}&\mathrm{SE}\big(\hat{\mathbf{X}}_k, \big[\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n_k)}\big]\big)\\ &\quad= \Big(\big(\hat{\mathbf{x}}_k^{(1)} - \mathbf{x}_k^{(1)}\big)^2 + \cdots + \big(\hat{\mathbf{x}}_k^{(n_k)} - \mathbf{x}_k^{(n_k)}\big)^2\Big).\end{aligned} \tag{8}$$

To evaluate the measure, we need estimates of the states of target 1, 2, and so on, i.e., of the labeled targets. Since this paper is about describing where there are targets, rather than where a target with a certain label is, the squared error is not a good measure. We note for example in the two-target case, that if the identities of the targets have been mixed up, the squared error can be very large, even though there might be two accurate tracks available, i.e., even though $\hat{\mathbf{x}}_k^{(1)} \approx \mathbf{x}_k^{(2)}$ and $\hat{\mathbf{x}}_k^{(2)} \approx \mathbf{x}_k^{(1)}$. Therefore, we seek a measure that can capture the quality of an algorithm to estimate the set of targets.

Several multi-target performance measures have been proposed in the literature. An ad-hoc optimal assignment-based approach, with arbitrary cost function, was given in [14], while the first rigorous theory of multi-object distances was given in [15]. The measure, which is based on the optimal assignment approach, is called the Optimal Mass Transfer (OMAT)

metric. When the number of targets is known, the OMAT metric is identical to the optimal assignment procedure. OMAT has a number of weaknesses, discussed in [7]. Apart from not being a metric, the major weaknesses appear when the number of targets is not known and the estimated number is not always equal to the true number. As a remedy, a new measure called the Optimal Subpattern Assignment (OSPA) metric, was proposed in [7]. Since OSPA both is a true metric and an intuitively appealing measure which has received much attention of late, we use it as the basis measure for the problem.

Let $\mathbf{X}$ be the set of true target states and $\widehat{\mathbf{X}}$ be the set of target estimates, in our case both with n elements. The OSPA measure $\tilde{d}_p^{(c)}$ is then defined as

$$\tilde{d}_p^{(c)}(\widehat{\mathbf{X}}, \mathbf{X}) = \left(\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^{n} d^{(c)}(\mathbf{x}^{(i)}, \hat{\mathbf{x}}^{\pi(i)})^p \right) \right)^{1/p}. \tag{9}$$

Here, $d^{(c)}(\mathbf{x}, \hat{\mathbf{x}}) \triangleq \min(c, d(\mathbf{x}, \hat{\mathbf{x}}))$ is the distance d between $\mathbf{x}$ and $\hat{\mathbf{x}}$, cut-off at c. Further, Π_n is the set of all possible permutations of $\widehat{\mathbf{X}}$. The notation $\hat{\mathbf{x}}^{\pi(i)}$ describes the i^{th} permutation (re-ordering) of the vector $\hat{\mathbf{x}}$. In this article, we let d be the Euclidean distance, and we use a quadratic measure ($p = 2$). In practice, the measure performs an optimal assignment of target estimates to true target states, possibly clamped at c.[1]

To describe the performance of an estimator, and to have a measure for which we can define an optimal algorithm, we average over all state vectors, which gives us a definition of the *mean* OSPA (MOSPA) measure

$$\text{MOSPA}_p^{(c)}(\widehat{\mathbf{X}}) \triangleq \mathbb{E}_{p(\mathbf{X}|\mathbf{Z}^k)}\{\tilde{d}_p^{(c)}\}. \tag{10}$$

An optimal estimator, in the MOSPA sense, is an estimator which minimizes the MOSPA measure. Such an estimator is referred to as a minimum MOSPA (MMOSPA) estimator[2].

2.3 Motivation of the problem and the MOSPA measure

The problem that we study in this paper is the problem of estimating the unordered set of targets, for which the MMOSPA estimator is optimal. We

[1]For $c = \infty$, the above measure $\tilde{d}_p^{(c)}$ is equal to the OMAT metric. A consequence of this relation, using results from [15], is that the OSPA measure for known target numbers, and $c = \infty$, reduces to the optimal assignment approach, presented earlier.

[2]Note that the relation between OSPA, MOSPA and MMOSPA is analogous to the relation between the common acronyms SE, MSE and MMSE.

here give three examples of when minimizing MOSPA provides a solution which is more reasonable than what is obtained when minimizing the Mean Square Error (MSE), using an ordered density.

The first application is in radar cueing, i.e., for the problem of steering a radar sensor to areas of high target existence probability. In such cases, there is no interest in which target is which, but the question of where there are targets is very important. The risk with using an ordered density for this application is that if there is an uncertainty in the labeling of the targets, the posterior density will be multimodal. Then, there is a high probability that the mean value is in an area of low target existence probability. Steering the main lobe of the sensor to such an area is thus prone to low probability of true-target returns. By disregarding ordering, the posterior density can be made less multimodal, and the steering of the main lobe can then with higher probability be directed to an area where targets are likely to be.

A second example is in the automotive industry: In collision avoidance systems, it is not of interest to know which car is which—the only interest is to avoid all cars. Finally, when tracking extended objects using radar measurements, those objects are often described by a set of reflectors. The problem of interest is then to track that set of reflectors, and not try to distinguish which reflector is which.

3 Conceptual solutions and the JPDA approximation

For the case of target tracking, there are two well-studied optimal, or conceptual, solutions, and these will be discussed in this section. By presenting these conceptual solutions, we believe that it is easier to understand the new approach of this paper, which is based on a third conceptual solution introduced in Section 4.

The basis of both conventional solutions is first to calculate an optimal description of the joint target density, and then to derive MMOSPA estimates from the optimal description. In the description of the first conceptual solution, we also describe the data association problem, and how it is solved in the conventional frameworks. At the end, we describe the JPDA approximation to the first conceptual solution.

3.1 Conceptual solution I – ordered densities

The first conceptual solution is to use the traditional approach of first calculating the ordered posterior density $p\big(\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n_k)} \big| \mathbf{Z}^k\big)$, and then to derive

MMOSPA state estimates from it. For the conventional problem of minimizing MSE, the traditional approach is the foundation for methods such as JPDA and MHT. The first conceptual solution to the considered problem is presented in Table 1.

Table 1: Conceptual solution I – ordered densities

1)	Compute the ordered density $p\big(\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n)} \big| \mathbf{Z}^k\big)$ recursively.
2)	Derive MMOSPA state estimates from the ordered density.

For an ordered density with known number of targets, n, the state vectors of the targets can be stacked in a long vector $\mathbf{X}_k$ as shown in (1). To compute MMOSPA estimates, we seek to calculate the posterior density $p\big(\mathbf{X}_k \big| \mathbf{Z}^k\big)$. By marginalizing over all global data association hypotheses, $\mathcal{H}$, the density is given by

$$p\big(\mathbf{X}_k \big| \mathbf{Z}^k\big) = \sum_{h \in \mathcal{H}} p\big(\mathbf{X}_k \big| h, \mathbf{Z}^k\big) \Pr\big\{h \big| \mathbf{Z}^k\big\} \tag{11}$$

where the densities $p\big(\mathbf{X}_k \big| h, \mathbf{Z}^k\big)$ are easy to express. A global data association hypothesis, h, determines which measurements are clutter and which are target-generated, and in the latter case their target of origin. If the process and measurement models are linear and Gaussian, the posterior density is a Gaussian mixture, with increasingly many components over time. Thus, the optimal data association solution is practically infeasible, and sub-optimal solutions are required. One such solution is given by the JPDA filter, which we describe in Section 3.3.

Apart from being computationally infeasible, there is also a second difficulty with the conceptual solution, namely that it is not obvious how estimates with low MOSPA should be derived from the ordered density. As we show in Example 1 in Section 4.2, the posterior mean, which is normally the estimate used in the traditional problem of minimizing MSE, might not at all be a suitable estimate in MOSPA sense.

3.2 Conceptual solution II – unordered densities

The second conceptual solution relies on the fact that to compute MMOSPA estimates, it is sufficient to know the unordered posterior density $p\big(\{\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n)}\} \big| \mathbf{Z}^k\big)$, since the labeling of the targets has no influence on the MOSPA measure. An optimal solution utilizing the sufficiency of the unordered density is given in Table 2.

Table 2: Conceptual solution II – unordered densities

1)	Compute the unordered density $p\big(\{\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n)}\}\big|\mathbf{Z}^k\big)$ recursively.
2)	Derive MMOSPA state estimates from the unordered density.

The second conceptual solution has the same two difficulties as the first one, viz. that approximations are necessary to keep complexity at a constant level and that derivation of MMOSPA estimates from an unordered density is a research topic in itself.

Step one of the second conceptual solution (cf. Table 2) is the foundation of the family of PHD/CPHD algorithms [9–12] with the Gaussian-mixture CPHD filter [12] as perhaps the most prominent one.

3.3 JPDA approximation

The JPDA filter is an approximative solution to the data association problem, described in the spirit of the first conceptual solution. The approach of the filter is to recursively approximate the multi-modal posterior density in (11) by a single Gaussian. The posterior density can be rewritten as

$$p\big(\mathbf{X}_k\big|\mathbf{Z}^k\big) = \sum_{h\in\mathcal{H}} \frac{p\big(\mathbf{Z}_k\big|\mathbf{X}_k, h, \mathbf{Z}^{k-1}\big)p\big(\mathbf{X}_k\big|\mathbf{Z}^{k-1}\big)}{p\big(\mathbf{Z}_k\big|\mathbf{Z}^{k-1}\big)} \times \Pr\big\{h\big|\mathbf{Z}^{k-1}\big\}. \tag{12}$$

Assuming linear and Gaussian models, and a Gaussian prior density $p\big(\mathbf{X}_k\big|\mathbf{Z}^{k-1}\big)$ (given by the JPDA approximation at time index $k-1$), the above density is a Gaussian mixture.

The JPDA filter performs the following steps:

1. Formulate all global data association hypotheses, $\mathcal{H}$, which describe possible origins of $\mathbf{Z}_k$.

2. For each data association hypothesis $h \in \mathcal{H}$, update the predicted density for each target i with the assigned measurement j as a Kalman filter update. The output of the filter is the mean value $\mathbf{x}_{k|k}^{i,h}$ and covariance matrix $\mathbf{P}_{k|k}^{i,h}$ (cf. (18)–(19)).

3. Calculate the weight β_h of each mixture component h.

4. Use moment matching to approximate the Gaussian mixture by a single Gaussian, see (17).

The mixture weights β_h are given by

$$\beta_h = \frac{\bar{\beta}_h}{\sum_{h=1}^{N_{\mathcal{H}}} \bar{\beta}_h} \tag{13}$$

$$\bar{\beta}_h = \prod_{\mathcal{S}_0^h} \lambda \cdot \prod_{\mathcal{S}_u^h} \left(1 - P_D\right) \cdot \prod_{\{i,j\} \in \mathcal{S}_a^h} P_D \, g_{ij}, \tag{14}$$

where a constant detection probability is assumed, $\mathcal{S}_0^h$ is the set of unassigned measurements, $\mathcal{S}_u^h$ is the set of unassigned targets, and $\mathcal{S}_a^h$ is a set including the pairs of detected targets, i, and their assigned measurements, j. Further,

$$g_{ij} = \mathcal{N}\big(\boldsymbol{\nu}_{ij}; \mathbf{0}, \mathbf{S}_k\big) = \frac{1}{\left|2\pi \mathbf{S}_k\right|^{1/2}} e^{-\frac{d_{ij}^2}{2}} \tag{15}$$

$$d_{ij}^2 = \boldsymbol{\nu}_{ij}^T \mathbf{S}_k^{-1} \boldsymbol{\nu}_{ij}, \quad \boldsymbol{\nu}_{ij} = \mathbf{z}_k^j - \mathbf{H}_k \mathbf{x}_{k|k-1}^{(i)}. \tag{16}$$

As described above, the first three steps of the JPDA algorithm describe the calculation of the components in the Gaussian mixture, while the final step is to approximate that Gaussian mixture density for each target with a single Gaussian. This is done by moment matching, i.e.,

$$p\big(\mathbf{x}_k^{(i)} \big| \mathbf{Z}^k\big) \cong \mathcal{N}\big(\mathbf{x}_k^{(i)}; \mathbf{x}_{k|k}^i, \mathbf{P}_{k|k}^i\big) \tag{17}$$

where[3]

$$\mathbf{x}_{k|k}^i = \sum_{h=1}^{N_{\mathcal{H}}} \beta_h \mathbf{x}_{k|k}^{i,h} \tag{18}$$

$$\mathbf{P}_{k|k}^t = \sum_{h=1}^{N_{\mathcal{H}}} \beta_h \left\{ \mathbf{P}_{k|k}^{i,h} + \big(\mathbf{x}_{k|k}^i - \mathbf{x}_{k|k}^{i,h}\big)\big(\mathbf{x}_{k|k}^i - \mathbf{x}_{k|k}^{i,h}\big)^T \right\} \tag{19}$$

and where $N_{\mathcal{H}}$ is the total number of hypotheses.

The JPDA algorithm is often described in an alternative, but equivalent, fashion [5], where the computation of the state estimates includes the calculation of a weighted measurement residual which is used in an ordinary Kalman filter update.

[3]Although the sums in (18) and (19) can be done as written, in practice there would be a step of marginalization over the single-target association events.

4 Conceptual solution III – ordered densities with switching

4.1 The RFS family

In this section we discuss the relationship between ordered and unordered densities, and the effects and possibilities of that relationship. Further, we introduce a new, third, conceptual solution to the problem of interest, and illustrate its properties using simple examples.

The first two conceptual solutions included the calculation of the ordered, and unordered, posterior densities of the joint target states, respectively. A key insight, upon which we capitalize, is that there is a relation between these densities. For n targets, the relation is the following

$$\begin{aligned} &p(\{\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n)}\} = \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n\}) \\ &\quad = \sum_{i=1}^{n!} p(\mathbf{x}_k^{(1)} = \boldsymbol{\alpha}_{m_1^i}, \ldots, \mathbf{x}_k^{(n)} = \boldsymbol{\alpha}_{m_n^i}), \end{aligned} \tag{20}$$

where $[\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n]$ is a point in the joint target state space, and m_j^i, for $j = 1, \ldots, n$, is index j in permutation number i. To go from an ordered density to a set density, we thus sum over all possible permutations of the state vector. For instance, when $n = 2$ a natural choice is to set $m_1^1 = 1$, $m_2^1 = 2$ and $m_1^2 = 2$ and $m_2^2 = 1$. One important consequence of this relation is described in the following proposition.

Proposition 1 *For $n > 1$, the mapping from densities of ordered state vectors, $p_i(\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n)})$, to RFS densities, $p(\{\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n)}\})$, is many-to-one.*

Since many ordered densities correspond to the same unordered density, and since the RFS density is sufficient to derive optimal estimates, it is fair to say that the ordered density contains more information than necessary. For all densities which correspond to the same unordered density, we make the following definition:

Definition 1 *When two labeled densities, $p_1(\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n)})$ and $p_2(\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n)})$, correspond to the same RFS density, we say that they belong to the same* RFS family*.*

So, using (20), we obtain the same RFS density regardless if $p_1(\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n)})$ or $p_2(\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n)})$ is used. In Fig. 1, two labeled densities in the same RFS

family are shown, together with the RFS density. Obviously, even though the ordered densities belong to the same RFS family, their shape, expected values and covariance matrices can be very different.

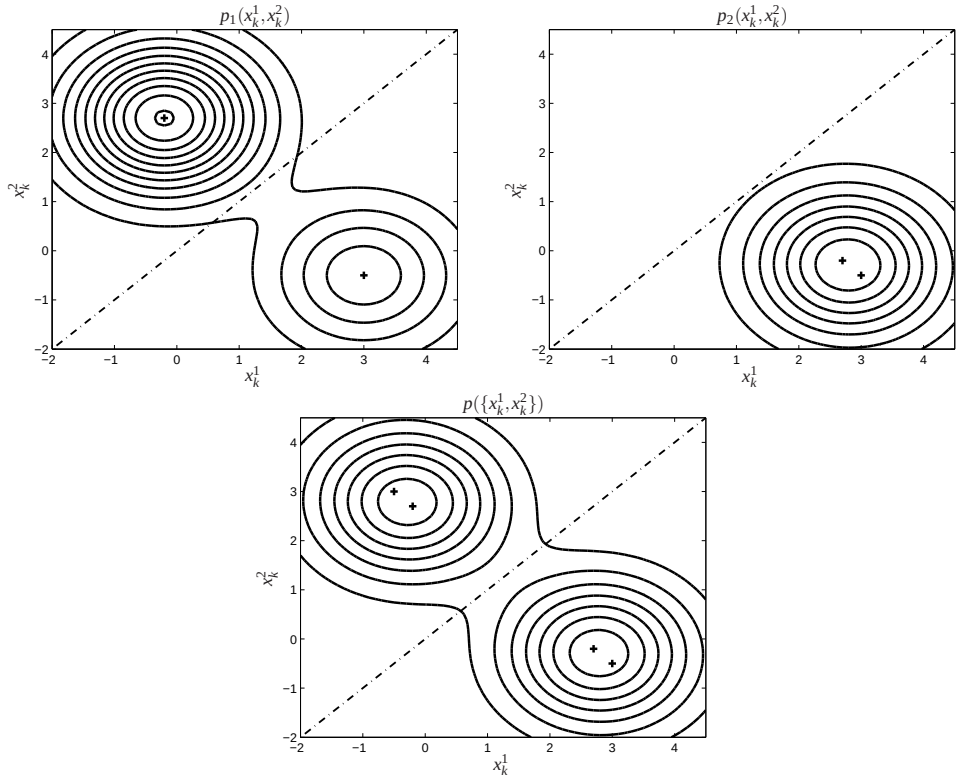

Figure 1: Marginalized posterior densities before [top left] and after [top right] switching the indexes under $\mathcal{H}_2$, and the RFS density [bottom]. In our opinion, a Gaussian approximation is much more appropriate in the middle figure than in the left one. The symmetry line is dash-dotted and the pluses indicate expected values of the Gaussian mixture components. The RFS density is inherently symmetric and is not affected by label switches in the ordered densities.

The MMOSPA estimate can be calculated from the RFS density, but also from the ordered densities. Since Proposition 1 tells us that several ordered densities correspond to the same RFS density, all these densities should result in the same MMOSPA estimates. This assumes, of course, that the estimates are computed using optimal algorithms. To find the optimal estimates, we can first calculate the RFS density and then find the MMOSPA estimate from it using optimal algorithms. All ordered densities which correspond to the same RFS density must hence yield the same MMOSPA estimate. We

summarize this in a proposition.

Proposition 2 *All ordered densities within an RFS family yield the same MMOSPA estimate. Further, that MMOSPA estimate is equal to the MMOSPA estimate obtained from the corresponding unordered density.*

So if we have the RFS family at a certain time, we can use any of the densities in the family to find the MMOSPA estimate, or we can use the corresponding RFS density. But what happens when we update the ordered densities with the same data? Will they still correspond to the same RFS family? With the following proposition, and its proof, we show that it is actually so.

Proposition 3 *Suppose* $p_1(\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)}, \ldots, \mathbf{x}_k^{(n)})$ *and* $p_2(\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)}, \ldots, \mathbf{x}_k^{(n)})$ *are two labeled densities within the same RFS family*[4]*. When these densities are updated using the same set of measurements,* $\mathbf{Z}_k$*, the updated densities still belong to the same RFS family.*

Proof of proposition 3 *The key to this result is the fact that given the state vector, the likelihood is the same for all terms in* (20)*, i.e., for all permutations of the target positions. Intuitively, this means that once the information about the target identities is lost, it can not be recovered from future data. Consequently, after the Bayesian update we get* $p_1(\{\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n)}\} = \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n\} \big| \mathbf{Z}_k)$

$$
\begin{aligned}
&= \sum_{i=1}^{n!} \left[\frac{p(\mathbf{Z}_k \big| \mathbf{x}_k^{(1)} = \boldsymbol{\alpha}_{m_1^i}, \ldots, \mathbf{x}_k^{(n)} = \boldsymbol{\alpha}_{m_n^i})}{p(\mathbf{Z}_k)} \right. \\
&\qquad \left. \times\, p_1(\mathbf{x}_k^{(1)} = \boldsymbol{\alpha}_{m_1^i}, \ldots, \mathbf{x}_k^{(n)} = \boldsymbol{\alpha}_{m_n^{(i)}}) \right] \qquad (21) \\
&= \frac{p(\mathbf{Z}_k \big| \mathbf{x}_k^{(1)} = \boldsymbol{\alpha}_1, \ldots, \mathbf{x}_k^{(n)} = \boldsymbol{\alpha}_n)}{p(\mathbf{Z}_k)} \\
&\qquad \times \sum_{i=1}^{n!} p_1(\mathbf{x}_k^{(1)} = \boldsymbol{\alpha}_{m_1^i}, \ldots, \mathbf{x}_k^{(n)} = \boldsymbol{\alpha}_{m_n^i}) \qquad (22) \\
&= p_2(\{\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n)}\} = \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n\} \big| \mathbf{Z}_k), \qquad (23)
\end{aligned}
$$

[4]We also assume that all targets have identical properties conditioned on the state vector, i.e., that the measurement equations are the same, independent of the target number. If this is not so, we have no business using the unordered densities in the first place.

where the last equality is due to $p_1(\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)}, \ldots, \mathbf{x}_k^{(n)})$ *and* $p_2(\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)}, \ldots, \mathbf{x}_k^{(n)})$ *being part of the same RFS family.*

As we can see, all labeled densities that correspond to the same RFS density, $\frac{1}{n!}\sum_{i=1}^{n!} p(\mathbf{x}_k^1 = \boldsymbol{\alpha}_{m_1^i}, \ldots, \mathbf{x}_k^n = \boldsymbol{\alpha}_{m_n^i})$ before measurement update, still belong to the same (only updated) RFS density, $p(\{\mathbf{x}_k^1 = \boldsymbol{\alpha}_1, \ldots, \mathbf{x}_k^n = \boldsymbol{\alpha}_n\}\big|\mathbf{Z}_k)$, after the measurement update.

To phrase this differently, all densities that belong to the same RFS family should result in the same MMOSPA estimates, both now and for all future times, i.e., also when new data is available. We can therefore replace our density at hand with any other density within the RFS family, without influencing the estimates[5]. We conclude our findings in a theorem.

Theorem 1 *In a recursive filtering framework, at each time instant we have the possibility of switching between ordered densities in the RFS family without affecting current or future MMOSPA estimates.*

From Theorem 1, we see that there is a previously unrecognized possibility of switching densities at any time in a filtering framework, without affecting the optimal performance. Based on this, we can describe a third conceptual solution (see Table 3) to the problem of this paper—a conceptual solution that utilizes the possibility of switching densities within the RFS family.

Table 3: Conceptual solution III – ordered densities with switching

1)	Compute the ordered density $p\big(\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n_k)}\big|\mathbf{Z}^k\big)$ recursively.
2)	Replace $p\big(\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n_k)}\big|\mathbf{Z}^k\big)$ with another ordered density which corresponds to the same RFS density.
3)	Derive MMOSPA state estimates from the ordered density.

This conceptual solution provides us with new possibilities and tools for designing novel suboptimal algorithms with better performance. As a switch within the RFS family does not affect the estimates, we could for instance make a change to a density which is more accurately approximated with a Gaussian density. The relevance of that is obvious, considering for example the JPDA filter, which relies heavily on Gaussian approximations. Since

[5]It is important, to go between (21) and (22), as with other steps in our development, that the targets be truly exchangeable and indistinguishable. For example, if target 1 has a different RCS than target 2, then this cannot be claimed.

computational complexity must be kept at a constant level, all practical algorithms need approximations, and Gaussian approximations are common. In the RFS family, there may be a density which much more resembles a Gaussian density. An example of that is seen in Fig. 1. By switching to that density, a more accurate Gaussian approximation is possible.

In practice, the posterior mean is often used as estimate also when the cost function is not the squared error. With the same reasoning as above, there may thus also exist densities within the RFS family whose expected value is closer to the optimal estimate, than the expected value of the original density. In fact, in [13] it is shown that there is a density in the RFS family whose mean value is the MMOSPA estimate. In Section 4.2, we study an example which shows that a density switch can lead to both improved approximations and to improved state estimates when the posterior mean is the estimator.

4.2 An example of the use of an RFS family

By considering an example, we here illustrate and discuss the potential benefits of density switches, both in terms of accuracy of the Gaussian approximations and the accuracy of the state estimates, when the posterior mean is used as estimate.

Example 1: Consider a scenario where we have two Gaussian distributed targets. The example is illustrated in Fig. 1. The probability of detection is one for both targets, and we have received two detections. We represent the two possible data associations by the hypotheses $\mathcal{H}_1$ and $\mathcal{H}_2$. Now, suppose that calculations yield the numbers[6],

$$\Pr\{\mathcal{H}_1\} = 0.3, \quad \mathbf{x}_k|\mathcal{H}_1 \sim \mathcal{N}\left(\begin{bmatrix} 3 \\ -0.5 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) \tag{24}$$

$$\Pr\{\mathcal{H}_2\} = 0.7, \quad \mathbf{x}_k|\mathcal{H}_2 \sim \mathcal{N}\left(\begin{bmatrix} -0.2 \\ 2.7 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right). \tag{25}$$

We realize that the two sets $\{x_k^1 = \beta_1, x_k^2 = \beta_2\}$ and $\{x_k^1 = \beta_2, x_k^2 = \beta_1\}$ represent the same set of targets. We may therefore move density from one such labeled point to the other, without changing the RFS density. One way to do this is by switching the indexes under $\mathcal{H}_2$,

$$\Pr\{\mathcal{H}_1\} = 0.3, \quad \mathbf{x}_k|\mathcal{H}_1 \sim \mathcal{N}\left(\begin{bmatrix} 3 \\ -0.5 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) \tag{26}$$

$$\Pr\{\mathcal{H}_2\} = 0.7, \quad \mathbf{x}_k|\mathcal{H}_2 \sim \mathcal{N}\left(\begin{bmatrix} 2.7 \\ -0.2 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right). \tag{27}$$

[6]All probabilities and densities are conditioned on data, but this is omitted for notational convenience.

The densities $p_1(\mathbf{x}_k)$, given by (24)–(25), and $p_2(\mathbf{x}_k)$, given by (26)–(27), thus correspond to the same RFS density

$$p(\{x_k^1, x_k^2\}) = p_1(\mathbf{x}_k) + p_1(\pi\mathbf{x}_k) = p_2(\mathbf{x}_k) + p_2(\pi\mathbf{x}_k), \tag{28}$$

where the matrix π is a permutation matrix, defined as

$$\pi = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{29}$$

Since the ordered densities p_1 and p_2 belong to the same RFS family, they should, ideally, render the same estimates. As illustrated in Fig. 1, the suggested switch leads to a simpler problem, since *the marginalized densities of $x_k^{(1)}$ and $x_k^{(2)}$ can be approximated by a Gaussian density more accurately.*

The dash-dotted line in Fig. 1 is a symmetry line. When probability mass is moved from one labeled point to another, the movement is through this line to the mirror point. Of course, the example is selected to highlight the advantages with a switch, and one can easily construct situations when it is better not to switch indices. Still, the example illustrates a general technique that can be employed by most tracking algorithms that use merging.

To further improve the understanding of the concepts in Example 1, we stress the relation to the RFS densities. Let $\tilde{p}_1(x_k^{(1)}, x_k^{(2)})$ and $\tilde{p}_2(x_k^{(1)}, x_k^{(2)})$ denote the Gaussian approximations of $p_1(x_k^{(1)}, x_k^{(2)})$ and $p_2(x_k^{(1)}, x_k^{(2)})$, respectively. Both $p_1(x_k^{(1)}, x_k^{(2)})$ and $p_2(x_k^{(1)}, x_k^{(2)})$ correspond to the same RFS density, i.e., $p_1(\{x_k^{(1)}, x_k^{(2)}\}) = p_2(\{x_k^{(1)}, x_k^{(2)}\})$ (cf. (28)). As the approximation $\tilde{p}_2(x_k^{(1)}, x_k^{(2)}) \approx p_2(x_1, x_k^{(2)})$ is fairly accurate, it follows that $\tilde{p}_2(\{x_k^{(1)}, x_k^{(2)}\}) \approx p_2(\{x_k^{(1)}, x_k^{(2)}\}) = p_1(\{x_k^{(1)}, x_k^{(2)}\})$. However, it is likely that the approximation $\tilde{p}_1(\{x_k^{(1)}, x_k^{(2)}\}) \approx p_1(\{x_k^{(1)}, x_k^{(2)}\})$ is less accurate. Hence, by switching densities we will make approximations that better preserve the information about the desired RFS density, $p_1(\{x_k^{(1)}, x_k^{(2)}\})$.

It is not obvious how to compute MMOSPA estimates from a given RFS density. Instead, the proposed algorithms in Sections 5 and 6 use the MMSE estimates, i.e., the posterior means, of a density of ordered targets. The idea is to select a density within the RFS family such that the MMSE estimates are close to the MMOSPA estimates. For the considered example, we illustrate the importance that the choice of density has on the posterior means and the MOSPA performance, by studying the MMSE state vector estimates before and after the index switch. In the original indexation, the posterior means are

$$\hat{x}_k^{(1)} = 0.76, \quad \hat{x}_k^{(2)} = 1.74 \tag{30}$$

whereas the posterior means after the switch are

$$\hat{x}_k^{(1)} = 2.79, \quad \hat{x}_k^{(2)} = -0.29. \tag{31}$$

The latter posterior means are probably close to the optimal estimates since, under both hypotheses, one target is fairly close to 2.79 whereas the other target is reasonably close to -0.29. Hence, although the initial objective was to improve the Gaussian approximation, the density switch also seems to yield MMSE estimates which are closer to the MMOSPA estimates.

For more results on the connection between accurate Gaussian approximations and improved estimates, we refer to Section 6.

5 Optimal approximations in the Kullback-Leibler sense

In this section, we present a way of switching densities within the RFS family such that the new density can be most accurately approximated with a Gaussian density, in the Kullback-Leibler sense. Note that the description is only made for a two-target case, but the results can be generalized.

In what follows, we assume that the posterior density is a Gaussian mixture, denoted

$$p(\mathbf{x}) = \sum_{h=1}^{N_{\mathcal{H}}} \beta_h \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_h, \mathbf{P}_h). \tag{32}$$

For the two-target case, we make the following definition of the RFS family, $\mathcal{A}_p$, of $p(\mathbf{x})$ (cf. (28))

$$\mathcal{A}_p \triangleq \{f : f(\mathbf{x}) + f(\pi\mathbf{x}) = p(\mathbf{x}) + p(\pi\mathbf{x})\}. \tag{33}$$

Here, $\mathbf{x}$ is the stacked vector of the state vectors $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ of the two targets, and π is a permutation matrix (cf. (29)).

Depending on the choice of density $f(\mathbf{x})$, the Gaussian approximation may be more or less accurate. Our objective here is to

1. find a density $f(\mathbf{x})$ that enables the most accurate Gaussian approximations; and then to

2. find the Gaussian density $\mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \mathbf{R})$ that best approximates $f(\mathbf{x})$.

As a measure for how accurate a density approximation is, we will use the Kullback-Leibler divergence [16]. Of course, our interest in $f(\mathbf{x})$ is only as a means to obtain $\mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \mathbf{R})$, which means that one could imagine a technique that does not involve $f(\mathbf{x})$. The mathematical definition of the problem that we would like to solve is

$$\{\bar{\mathbf{x}}, \mathbf{R}\} = \arg\min_{\bar{\mathbf{x}}, \mathbf{R}} \left\{ \min_{f(\mathbf{x}) \in \mathcal{A}_p} \mathrm{KL}\left[f(\mathbf{x}) || \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \mathbf{R}) \right] \right\}, \tag{34}$$

where

$$\mathrm{KL}\left[f(\mathbf{x}) || \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \mathbf{R}) \right] \triangleq \int f(\mathbf{x}) \log \frac{f(\mathbf{x})}{\mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \mathbf{R})} \, d\mathbf{x}, \tag{35}$$

and where KL denotes Kullback-Leibler divergence. In the standard procedure (used in for instance PDA and JPDA) one would use $f(\mathbf{x}) = p(\mathbf{x})$ and only optimize over $\bar{\mathbf{x}}$ and $\mathbf{R}$, for which the optimal solution is given by moment matching. By also optimizing over $f(\mathbf{x})$ we believe that the approximation errors will decrease significantly compared to JPDA, and slightly compared to the algorithm that we present in Section 6.

To design an algorithm that can search for $\bar{\mathbf{x}}$ and $\mathbf{R}$, the following results are very useful.

Theorem 2 *The solution to*

$$\{\bar{\mathbf{x}}, \mathbf{R}\} = \arg\min_{\bar{\mathbf{x}}, \mathbf{R}} KL\left[f(\mathbf{x}) || \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \mathbf{R}) \right] \tag{36}$$

is given by moment matching,

$$\bar{\mathbf{x}} = \mathbb{E}_{f(\mathbf{x})} \{\mathbf{x}\} \tag{37}$$

$$\mathbf{R} = Cov_{f(\mathbf{x})} \{\mathbf{x}\}. \tag{38}$$

Furthermore, the density $f(\mathbf{x}) \in \mathcal{A}_p$ that minimizes $KL\left[f(\mathbf{x}) || \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \mathbf{R}) \right]$ is

$$f(\mathbf{x}) = (p(\mathbf{x}) + p(\pi\mathbf{x})) \cdot \frac{\mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \mathbf{R})}{\mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \mathbf{R}) + \mathcal{N}(\pi\mathbf{x}; \bar{\mathbf{x}}, \mathbf{R})}. \tag{39}$$

Proof of Theorem 2 *The results in Eq.* (37) *and* (38) *are well known, see e.g. [17]. For* (39)*, see [18].*

Based on the above theorem, we propose an iterative optimization algorithm. The algorithm is illustrated in Fig. 2, where the initial density is $p(\mathbf{x})$.

1. Initiate with $i = 1$ and let $\bar{\mathbf{x}}_0$ and $\mathbf{R}_0$ be the first two moments of $p(\mathbf{x})$.

2. Set

$$f_i(\mathbf{x}) = (p(\mathbf{x}) + p(\pi\mathbf{x})) \frac{\mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}_{i-1}, \mathbf{R}_{i-1})}{\mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}_{i-1}, \mathbf{R}_{i-1}) + \mathcal{N}(\pi\mathbf{x}; \bar{\mathbf{x}}_{i-1}, \mathbf{R}_{i-1})}. \tag{40}$$

3. Compute

$$\bar{\mathbf{x}}_i = \mathbb{E}_{f_i(\mathbf{x})}\{\mathbf{x}\} \tag{41}$$
$$\mathbf{R}_i = \mathrm{Cov}_{f_i(\mathbf{x})}\{\mathbf{x}\}. \tag{42}$$

4. If $\bar{\mathbf{x}}_i \approx \bar{\mathbf{x}}_{i-1}$ and $\mathbf{R}_i \approx \mathbf{R}_{i-1}$ we stop. Otherwise, set $i = i + 1$ and go back to 2.

The algorithm is essentially straightforward. However, to implement it we need the ability to calculate expected values with respect to the densities $f_i(\mathbf{x})$, $i = 1, 2, \ldots$.

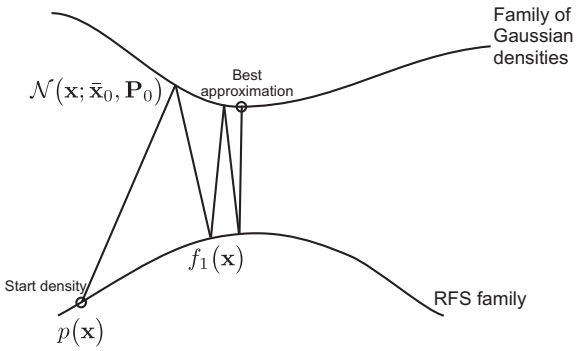

Figure 2: Illustration of the iterative optimization algorithm which finds the best Gaussian approximation of the RFS family, in the Kullback-Leibler sense. The curves represent the functional spaces of densities within the RFS family (below), and the Gaussian densities (above).

It appears complicated to find analytical expressions for the expected values in (41) and (42) and we therefore suggest a numerical method based on importance sampling. As importance density we use

$$q(\mathbf{x}) = \frac{1}{2}\big(p(\mathbf{x}) + p(\pi\mathbf{x})\big). \tag{43}$$

The expected values can be reformulated as

$$\bar{\mathbf{x}}_i = \mathbb{E}_{f_i(\mathbf{x})}\{\mathbf{x}\} = \mathbb{E}_{q(\mathbf{x})}\left\{\mathbf{x}\frac{f_i(\mathbf{x})}{q(\mathbf{x})}\right\} \tag{44}$$

$$\mathbf{R}_i = \mathbb{E}_{f_i(\mathbf{x})}\left\{(\mathbf{x}-\bar{\mathbf{x}}_i)(\mathbf{x}-\bar{\mathbf{x}}_i)^T\right\} \tag{45}$$

$$= \mathbb{E}_{q(\mathbf{x})}\left\{(\mathbf{x}-\bar{\mathbf{x}}_i)(\mathbf{x}-\bar{\mathbf{x}}_i)^T\frac{f_i(\mathbf{x})}{q(\mathbf{x})}\right\}, \tag{46}$$

and for the selected importance density it is easy to evaluate the ratio

$$\frac{f_i(\mathbf{x})}{q(\mathbf{x})} = 2\frac{\mathcal{N}(\mathbf{x};\bar{\mathbf{x}}_{i-1},\mathbf{R}_{i-1})}{\mathcal{N}(\mathbf{x};\bar{\mathbf{x}}_{i-1},\mathbf{R}_{i-1})+\mathcal{N}(\pi\mathbf{x};\bar{\mathbf{x}}_{i-1},\mathbf{R}_{i-1})}. \tag{47}$$

Based on i.i.d. samples $\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N \sim q(\mathbf{x})$, we can approximate the desired entities as

$$\bar{\mathbf{x}}_i \approx \sum_{n=1}^{N}\mathbf{x}_n w_n \tag{48}$$

$$\mathbf{R}_i \approx \sum_{n=1}^{N}(\mathbf{x}_n-\bar{\mathbf{x}}_i)(\mathbf{x}_n-\bar{\mathbf{x}}_i)^T w_n, \tag{49}$$

where

$$w_n = \frac{f_i(\mathbf{x}_n)/q(\mathbf{x}_n)}{\sum_{r=1}^{N} f_i(\mathbf{x}_r)/q(\mathbf{x}_r)}. \tag{50}$$

In (49), we replace $\bar{\mathbf{x}}_i$ with the approximated value from (48). A beneficial property with the suggested importance function is that since the RFS family is preserved we can use the same samples $\mathbf{x}_1,\ldots,\mathbf{x}_N$ for all iterations $i = 1,2,\ldots$.

In this section, we have presented the optimal approach of performing density switches within the RFS family, such that the final density can most accurately be approximated with a Gaussian density. We have also presented a numerical approach of computing the required expected values. Since the presented algorithm is computationally demanding, we are interested in a sub-optimal approach, which still has the properties of enabling better Gaussian approximations than the JPDA algorithm, and which also presents better estimates than JPDA in MOSPA sense. The development of such an approach is the topic of the following section.

6 Set JPDA algorithm

In this section, we derive a sub-optimal approach to the third conceptual solution, called the Set JPDA (SJPDA) algorithm. The filter utilizes the fact that several ordered densities correspond to the same set density, which provides the possibility to switch between those ordered densities. The criterion that we try to minimize with our switch of densities is optimal in terms of the MOSPA estimates. Still, we argue that it in most cases also gives better density approximations than the JPDA algorithm.

The Set JPDA algorithm is a modification of the classic JPDA filter. The difference is that once the posterior density is described as a weighted sum of Gaussian densities, we allow ourself to switch that density for another ordered density in the same RFS family. On a high level, the SJDA algorithm works as follows:

1) Formulate the set of global measurement hypotheses, $\mathcal{H}$, and calculate conditional densities of all targets as well as the probabilities of all hypotheses. Approximate the targets as independent.

2) Reorder the target indexes under the different hypotheses with the objective to make the marginalized densities resemble Gaussian densities (by minimization in one's favorite sense, see Sections 6.1 and 6.2 for details).

3) Approximate the marginalized posterior densities of all targets as independent Gaussian. Then go back to 1.

A block diagram description of the SJPDA filter is shown in Fig. 3. As seen in the figure, the difference between SJPDA and JPDA lies in the switching block. If that block is removed, we obtain the JPDA filter.

The key aspect of the SJPDA filter is the switching of densities. To obtain a filter with good performance, and which enables accurate Gaussian approximations, the switching criterion is very important. In the following two sections, we propose a goal function, and formulate the problem of minimizing that function (while remaining within the family of Gaussian mixtures) as a continuous optimization problem.

6.1 Goal function proposal and motivation

For the considered problem, the labeled density is a Gaussian mixture. In this section, we propose and motivate a goal function for the optimization problem of finding the best switching of such densities in the SJPDA filter.

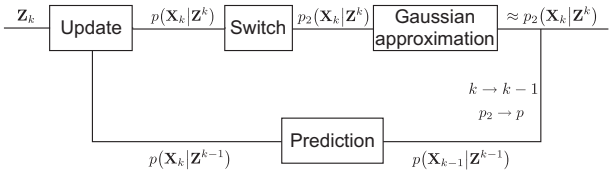

Figure 3: SJPDA block diagram.

Generally speaking, a Gaussian mixture can be accurately approximated by a single Gaussian density as long as the Gaussian mixture is not too distinctly multimodal. In the SJPDA filter, we wish to adjust the indexation within each global hypothesis, in order to find a labeled density which is less multimodal. The goal function that we propose is hence a function which measures the multi-modality of the density. The proposed goal function is

$$\sum_{i=1}^{n} \mathrm{tr}\{\mathbf{P}_k^i\}, \tag{51}$$

for which $\mathbf{P}_k^i$ is given in (19). In the following, we motivate that goal function, basically by arguing that through minimizing of the function, we will obtain both better approximations and better estimates than what we would obtain using the JPDA filter.

In [13], the problem of minimizing MOSPA is considered in more detail. An important result from the paper is that *the problem of minimizing the sum of the trace of the posterior covariances is equal to the problem of minimizing MOSPA*, given that the expected value is used as estimate and that we search in the entire RFS family (not restricting the search to Gaussian mixtures in the RFS family). Thus, the switching criterion that we use is optimal for the estimation problem.

The second advantage with the cost function is that it enables better Gaussian approximations. The reason for this is that by minimizing the trace of the covariance matrices, we make the posterior density less multimodal. We therefore argue that the Gaussian approximations in the SJPDA filter are at least as accurate as the JPDA filter approximations, where equality holds if no switches are made. However, for the density approximations, there are no optimality results. That is, one cannot show that the approximations are optimal, for example in a Kullback-Leibler sense, by using the proposed goal function. Instead, for optimal approximations, we refer to the algorithm given in Section 5.

6.2 SJPDA as a continuous optimization problem

In [19], the switching problem of SJPDA was formulated as a discrete optimization problem, for which a brute force solution can be implemented easily. In this section, we instead show how the problem can be reformulated as a continuous optimization problem, for which fast solvers can be applied.

We start by assuming that the posterior density is described by a Gaussian mixture, as in (32). To give the problem a continuous formulation, we introduce the variable $\phi_{i,h}$ which represent the weight of the Gaussian component after permutation. In this way, we can have a linear combination of the original component and the permuted component. For the discrete problem, $\phi_{i,h}$ is either 0 or β_h. In the continuous formulation of the problem, the posterior after switching is given by

$$q(\mathbf{x}) = \sum_{i=1}^{n!} \sum_{h=1}^{N_{\mathcal{H}}} \phi_{i,h} \mathcal{N}(\mathbf{x}; \pi_i \boldsymbol{\mu}_h, \pi_i \mathbf{P}_h \pi_i^T), \tag{52}$$

where π_i is permutation matrix number i, and where $n!$ is the total number of possible permutations. The new weights have to fulfill the following constraints

$$\sum_{i=1}^{n!} \phi_{i,h} = \beta_h \quad \forall h \tag{53}$$

$$\phi_{i,h} \geq 0 \quad \forall i, h. \tag{54}$$

Note that the densities $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_h, \mathbf{P}_h)$ and $\mathcal{N}(\mathbf{x}; \pi_i \boldsymbol{\mu}_h, \pi_i \mathbf{P}_h \pi_i^T)$ lie in the same RFS family for all permutation matrices π_i.

To conveniently express the goal function in the above parameters, we introduce the notations

$$\boldsymbol{\phi} = \begin{bmatrix} \phi_{1,1} & \dots & \phi_{n!,1} & \phi_{1,2} & \dots & \phi_{n!,2} & \dots & \phi_{n!,N_{\mathcal{H}}} \end{bmatrix}^T \tag{55}$$

$$\mathbf{V} = \begin{bmatrix} \pi_1 \boldsymbol{\mu}_1 & \dots & \pi_{n!} \boldsymbol{\mu}_1 & \pi_1 \boldsymbol{\mu}_2 & \dots & \pi_{n!} \boldsymbol{\mu}_2 \\ & & & \dots & & \pi_{n!} \boldsymbol{\mu}_{N_{\mathcal{H}}} \end{bmatrix}, \tag{56}$$

such that

$$\bar{\mathbf{x}} \triangleq \sum_{i=1}^{n!} \sum_{h=1}^{N_{\mathcal{H}}} \pi_i \boldsymbol{\mu}_h \phi_{i,h} = \mathbf{V} \boldsymbol{\phi}. \tag{57}$$

We wish to minimize

$$\begin{aligned}
&\mathrm{tr}\left\{\mathrm{Cov}_{q(\mathbf{x})}\left\{\mathbf{x}\right\}\right\} \\
&= \sum_{i=1}^{n!}\sum_{h=1}^{N_{\mathcal{H}}}\phi_{i,h}\Big\{(\pi_i\boldsymbol{\mu}_h-\bar{\mathbf{x}})^T(\pi_i\boldsymbol{\mu}_h-\bar{\mathbf{x}})+\mathrm{tr}\{\pi_i\mathbf{P}_h\pi_i^T\}\Big\} && (58)\\
&= \left\{\sum_{i=1}^{n!}\sum_{h=1}^{N_{\mathcal{H}}}\phi_{i,h}\boldsymbol{\mu}_h^T\boldsymbol{\mu}_h\right\} - \bar{\mathbf{x}}^T\bar{\mathbf{x}} + \sum_{h=1}^{N_{\mathcal{H}}}\mathrm{tr}\{\mathbf{P}_h\}\sum_{i=1}^{n!}\phi_{i,h} && (59)\\
&= \left\{\sum_{h=1}^{N_{\mathcal{H}}}\beta_h\boldsymbol{\mu}_h^T\boldsymbol{\mu}_h\right\} - \boldsymbol{\phi}^T\mathbf{V}^T\mathbf{V}\boldsymbol{\phi} + \sum_{h=1}^{N_{\mathcal{H}}}\mathrm{tr}\{\mathbf{P}_h\}\beta_h, && (60)
\end{aligned}$$

where $-\boldsymbol{\phi}^T\mathbf{V}^T\mathbf{V}\boldsymbol{\phi}$ is the only part that depends on $\boldsymbol{\phi}$, subject to the constraints presented in (53) and (54). To summarize, we have the optimization problem

$$\min_{\boldsymbol{\phi}} \quad -\boldsymbol{\phi}^T\mathbf{V}^T\mathbf{V}\boldsymbol{\phi}. \qquad (61)$$
$$\text{s.t.} \begin{cases} \phi_{i,h} \geq 0 & \forall i,h \\ \sum_{i=1}^{n!}\phi_{i,h} = \beta_h & \forall h \end{cases}$$

The problem is non-convex. The goal function is concave, and it is to be minimized over a convex region, which has the implication that the optimal point is along the border of the constraints. We also note that since the optimal solution is on the border of the constraints, the permutations either occur or not, i.e., there will be no partial permutations in the optimal point. A benefit with the continuous formulation is that it enables the use of commercial optimization solvers, which scale well with increasing number of targets and hypotheses. To use an optimization solver, we need knowledge regarding existence of local minima, and how they can be avoided.

It can be analytically shown that there are no local minima for the case of scalar state vectors and two targets. But when the dimensionality of the state vector increases, there will in some situations exist local minimum points in which an optimization algorithm can be trapped. It is thus important to select a suitable starting point for the algorithm. Through empirical studies, we have seen that it is only in a small region around the minimum point that the negative gradient points in the direction of the local minimum. Therefore, only starting points in a small region around potential local minima should be avoided. For the problems considered in this article, we have found a suitable set of starting points which almost always lead the selected optimization algorithm to the global minimum.

For more details on the characteristics of the optimization problem, see [18].

6.3 Algorithm description

The SJPDA algorithm has five steps which are executed for each time index k. In the following, we describe each of those five steps in more detail, and summarize the algorithm in Table 4.

The algorithm description starts at time k, at which we have available a set of measurements, $\mathbf{Z}_k$, and predicted states $\mathbf{x}^i_{k|k-1}$ and covariance matrices $\mathbf{P}^i_{k|k-1}$ for each target i. Note that steps I, II and V are identical to the JPDA algorithm (one version of it). Note also that we assume linear and Gaussian process and measurement models in the description, although the SJPDA algorithm can be easily extended to handle nonlinear models using, for example, an extended Kalman filter (EKF).

Step I: Data hypothesis extraction

The first step of the SJPDA algorithm is to formulate all possible global data association hypotheses, $\mathcal{H}$. A hypothesis is possible if it describes the origin of each measurement in $\mathbf{Z}_k$ (target-generated or false), and if the total number of target-originated measurements is at most n.

Step II: Data update

The second step of the algorithm is to update the predicted state vectors and covariance matrices, and to calculate the mixture weights, β_h. For a certain target i, the updated state and covariance matrix under hypothesis h are given by

$$\mathbf{x}^{i,h}_{k|k} = \mathbf{x}^i_{k|k-1} + \mathbf{K}^i_k\big(\mathbf{H}_k\mathbf{x}^i_{k|k-1} - \mathbf{z}^{i,h}_k\big) \tag{62}$$

$$\mathbf{P}^{i,h}_{k|k} = \big(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\big)\mathbf{P}^i_{k|k-1}, \tag{63}$$

where $\mathbf{z}^{i,h}_k$ is the measurement associated to target i under hypothesis h. If no measurement is associated, we use $\mathbf{z}^{i,h}_k = \mathbf{0}$. An expression for the mixture weights is given in (13).

Step III: Optimization

Step three of the SJPDA algorithm is the main step, namely to find the optimal permutation of state vectors under the data association hypotheses. We find the optimum solution, $\boldsymbol{\phi}^*$, to (61) by applying an optimization solver

Table 4: SJPDA algorithm description

I: Data hypothesis extraction

formulate all $N_{\mathcal{H}}$ global data association hypotheses

II: Data update

for each data association hypothesis $h = 1, 2, \ldots, N_{\mathcal{H}}$, do

for each target $i = 1, \ldots, n$, do

update the expected value and covariance matrix of the predicted state vector $\hat{\mathbf{x}}_k^i$ according to (62)–(63)

calculate the mixture weight β_h according to (13)

III: Optimization

solve the problem defined in (61) using an optimization solver

a suitable pair of starting points is given in (64)–(65), and a useful third initiation point is to perform a 70% permutation under the N (e.g. 10) largest hypotheses

IV: Permutation and update

compute the expected value and covariance matrix of the joint state vector $\mathbf{X}_k$ after data update, optimal permutation and Gaussian approximation, according to (68)–(69)

V: Prediction

for each target $i = 1, 2, \ldots, N_t$, do

predict the state vector and covariance matrix of target i, according to (70)–(71).

with different initiation points. A suitable pair of starting points for the two-target problem is

$$\boldsymbol{\phi}_{\text{start},1} = \begin{bmatrix} 0.7\beta_1 & 0.3\beta_1 & \ldots & 0.7\beta_{N_h} & 0.3\beta_{N_h} \end{bmatrix}^T \tag{64}$$

$$\boldsymbol{\phi}_{\text{start},2} = \begin{bmatrix} 0.7\beta_1 & 0.3\beta_1 & 0.3\beta_2 & 0.7\beta_2 \\ & \ldots & 0.7\beta_{N_h} & 0.3\beta_{N_h} \end{bmatrix}^T. \tag{65}$$

Step IV: Permutation and update

After the optimal solution $\boldsymbol{\phi}^*$ has been found, we permute the state vectors according to that solution. The permutation does not have to be done explicitly, since we are only interested in the Gaussian approximation of the permuted posterior density. To express the mean and covariance after permutation, we describe the calculation in the joint state vector $\mathbf{X}_k$. Before permutation, the expected value and covariance matrix of that vector is

$$\mathbf{X}_{k|k}^h = \left[\left(\mathbf{x}_{k|k}^{1,h}\right)^T \left(\mathbf{x}_{k|k}^{2,h}\right)^T \dots \left(\mathbf{x}_{k|k}^{n,h}\right)^T \right]^T \tag{66}$$

$$\mathbf{P}_{k|k}^h = \operatorname{diag}\{\mathbf{P}_{k|k}^{1,h}, \mathbf{P}_{k|k}^{2,h}, \dots, \mathbf{P}_{k|k}^{n,h}\}. \tag{67}$$

After permutation, the expected value and covariance matrix are given by

$$\mathbf{X}_{k|k} = \mathbf{V}\boldsymbol{\phi}^* \tag{68}$$

$$\begin{aligned} \mathbf{P}_{k|k} = \sum_{h=1}^{N_h} \sum_{j=1}^{n!} \boldsymbol{\phi}^*\big(i(h,j)\big) \Big[& \boldsymbol{\Pi}_j \mathbf{P}_{k|k}^h \boldsymbol{\Pi}_j \\ & + \big(\boldsymbol{\Pi}_j \mathbf{X}_{k|k}^h - \mathbf{X}_{k|k}\big)\big(\boldsymbol{\Pi}_j \mathbf{X}_{k|k}^h - \mathbf{X}_{k|k}\big)^T \Big], \end{aligned} \tag{69}$$

where the function $i(h,j)$ gives the index in $\boldsymbol{\phi}^*$ which corresponds to permutation i and hypothesis h. We approximate the targets as independent, which implies that the posterior covariance matrix $\mathbf{P}_k^i$ for target i (the indices are irrelevant) is the corresponding block in the matrix $\mathbf{P}_{k|k}$.

Step V: Prediction

The final step of the SJPDA algorithm is to let $k \to k-1$, and predict the state and covariance matrices of the set of targets to the next time step k. For linear models, that prediction is done using the ordinary Kalman prediction equations

$$\mathbf{x}_{k|k-1}^i = \mathbf{F}_k \mathbf{x}_{k-1|k-1}^i \tag{70}$$

$$\mathbf{P}_{k|k-1}^i = \mathbf{F}_k \mathbf{P}_{k-1|k-1}^i \mathbf{F}_k^T + \mathbf{Q}_k. \tag{71}$$

In Fig. 4, the densities of the joint state vector (with two targets and scalar states) is illustrated for different steps in the JPDA and SJPDA algorithms. In the example, the predicted density is a Gaussian, given by the top figure. The second row of densities show the density after measurement update and optimal permutations (for SJPDA). From the figures, we see that the SJPDA density can be described much more accurately by a Gaussian density. The last row shows the Gaussian approximations of the JPDA

and SJPDA posterior densities. It is clear that the covariance matrix of the SJPDA approximation is smaller.

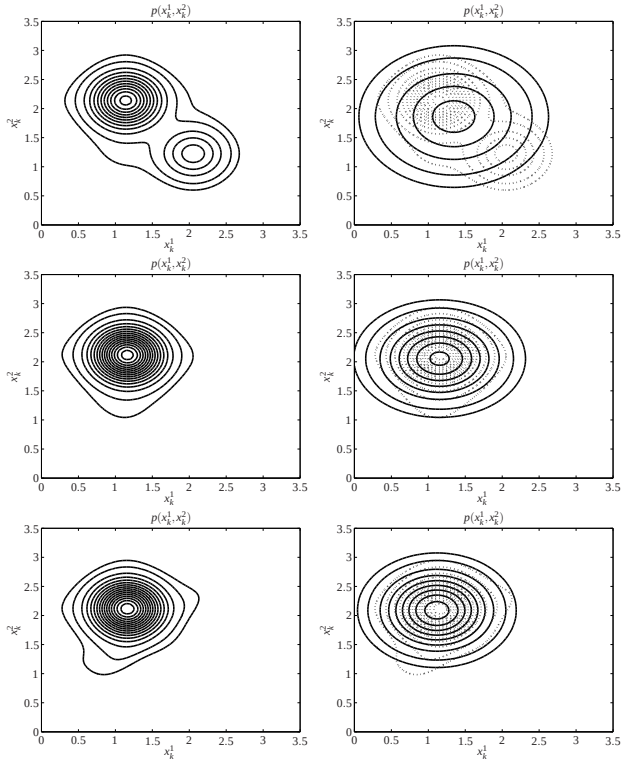

Figure 4: One-step iteration of the JPDA, SJPDA and KLSJPDA filters. The prior density is a Gaussian, there are two targets, the detection probability is 0.7, and two target-originated measurements are received. Posterior density [top left], posterior density after SJPDA optimization [middle left], posterior density after KLSJPDA optimization [bottom left], JPDA approximation [top right], SJPDA approximation [middle right], KLSJPDA approximation [bottom right]. The dashed contours in the right-column figures represent the densities that are approximated. Clearly, both the SJPDA and KLSJPDA posterior densities are better approximated by a Gaussian density than the original posterior density.

7 Evaluations

Two different tracking scenarios are considered for evaluation of the SJPDA algorithm, and comparison with JPDA and a Gaussian-mixture CPHD filter with known target number. The first scenario, for which also the KLSJPDA

algorithm is evaluated, is tracking of two targets, whose trajectories are as illustrated in Fig. 5. The parameters used in the evaluation are $l = 10$ m, $d_1 = 0.5$ m, $d_2 = 30$ m, $\varphi = \pi/3$, and $v = 1$ m/s. The second scenario, illustrated in Fig. 8, is also a two-target example, but in that scenario the two targets move around a hexagonal shape. The aim of that scenario is to evaluate how fast the JPDA and SJPDA algorithms lose track in a challenging scenario. In both scenarios, a sensor collects measurements at even time intervals of $T_0 = 1$ second.

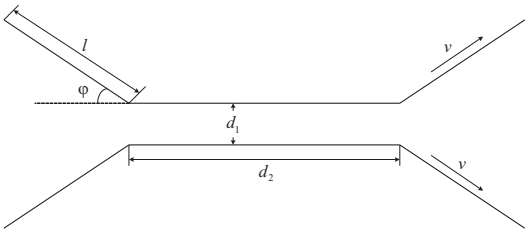

Figure 5: Illustration of the scenario for the first evaluation.

In the filtering algorithms, a nearly constant velocity model is assumed (see (4) for a general description), which is governed by the system matrix

$$\mathbf{F}_{k-1} = \begin{bmatrix} \mathbf{I}_{2\times 2} & T_0\mathbf{I}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \mathbf{I}_{2\times 2} \end{bmatrix}, \tag{72}$$

where $\mathbf{I}_{2\times 2}$ is a 2×2 identity matrix, and $\mathbf{0}_{2\times 2}$ is a 2×2 matrix of zeroes, and by the process noise $\mathbf{v}_{k-1}$ which is zero-mean Gaussian with covariance matrix

$$\mathbf{Q} = q_0 \begin{bmatrix} T_0^3\mathbf{I}_{2\times 2}/3 & T_0^2\mathbf{I}_{2\times 2}/2 \\ T_0^2\mathbf{I}_{2\times 2}/2 & T_0\mathbf{I}_{2\times 2} \end{bmatrix}, \tag{73}$$

where q_0 is a tuning parameter. In the simulations, the best tuning parameter is selected for each filter, where the parameter which yields the lowest average MOSPA is used. Further, the measurement model is assumed linear and Gaussian (cf. (6)) with observation matrix

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \tag{74}$$

which means that the sensor delivers position measurements. The Gaussian measurement noise $\mathbf{w}_k$ is zero-mean, with covariance matrix

$$\mathbf{R}_k = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}, \tag{75}$$

where we have used $\sigma_x = \sigma_y = 0.2$m.

Fig. 6 shows an example of the JPDA and SJPDA output for $d_1 = 0.5$m, $\lambda = 0.01$, and $P_D = 1$. In the figure we see the track coalescence tendency of the JPDA filter, which makes it hard for the filter to detect the separation of the tracks. This leads to high MOSPA after the track separation, and it also leads to a high risk of losing tracks. In Fig. 7, the MOSPA performance over 100 Monte Carlo runs is shown for detection probabilities of 1 and 0.85, respectively. The clutter intensity in the simulations is 0.02 m^{-2}.

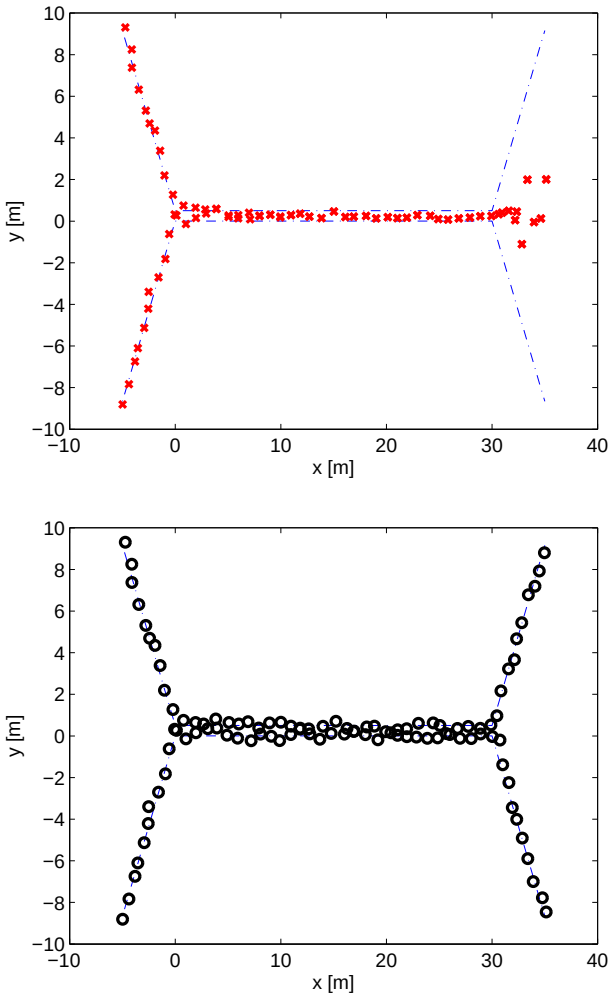

Figure 6: Output of JPDA [top] and SJPDA [bottom] for one sequence of measurements.

The figures show that the average OSPA performance of the SJPDA filter is better than for the JPDA filter for almost the entire scenario, and the

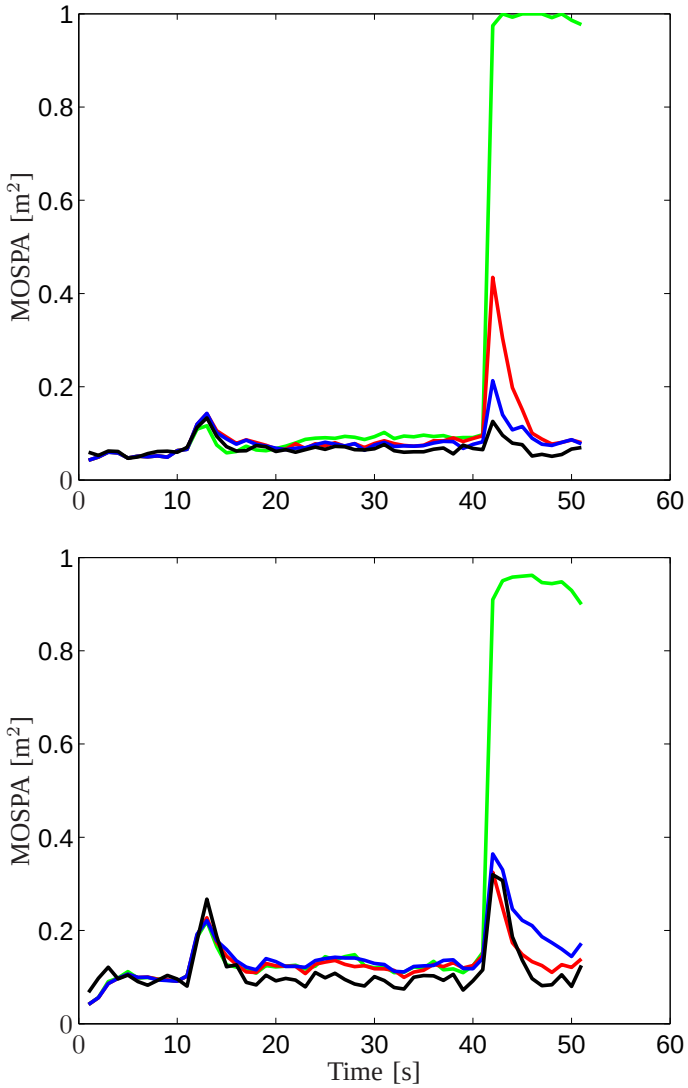

Figure 7: Evaluation of JPDA (green), SJPDA (red), KLSJPDA with 10 000 samples (blue), and GM-CPHD (black) on the two-target scenario for detection probabilities $P_d = 1$ [top] and $P_d = 0.85$ [bottom].

figures also show that the difference between the filters are very large at the time when the tracks separate. This is due to the track coalescence of the JPDA filter. Note that the filters are run with different process noises, where the noise level has been selected to yield the best average MOSPA. When the filters are run with the same process noise, the SJPDA filter always gives an average OSPA which is lower than, or equal to, that of the JPDA filter. For the KLSJPDA algorithm, the performance is similar to that of the SJPDA

filter, whereas its computational complexity is much higher. We thus suggest that the SJPDA filter be used.

When compared to GM-CPHD, we see that both the SJPDA and the KLSJPDA filters for large detection probabilities performs on average equally well as GM-CPHD (for unity detection probability actually slightly better), and almost as well for lower detection probabilities. The results are rather surprising since the GM-CPHD is a more complex algorithm. For the GM-CPHD algorithm, the best process noise parameter q_0 in the aforementioned set is used, and the filter also uses pruning (threshold 0.0001) and merging (threshold $U = 1$, cf. [12]). We note that the somewhat similar JPDA* [20] was compared to the SJPDA in [21]; in fact the JPDA* performs well, but the SJPDA is notably superior.

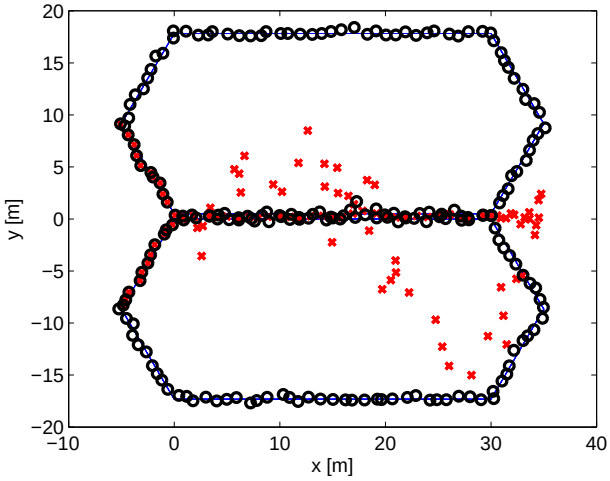

Figure 8: Example of filter output of the JPDA (crosses) and SJPDA (circles) algorithms for the track life evaluation scenario (for $P_D = 1$). In the plot, the output for the first 101 time steps is presented, which is the time it takes for the two targets to make one loop around the path. In the evaluations, the full trajectories make ten loops around the hexagonal path.

For the second scenario, the average track life and average track loss for JPDA and SJPDA are summarized in Table 5, for four different values of detection probability. In the evaluation, the best process noise parameter in the set $q_0 = \{0.05, 0.1, 0.2, 0.3, 0.4\}$ has been used, and otherwise the same parameter values as before. A track is considered lost if the covariance matrix elements corresponding to the uncertainty in the x or y dimensions have surpassed 225m^2, the estimate of a track is further away than 50 meters from the true value, or if there has been only two measurements in the gate

of a track for the last five scans. The selected process noise is such that the average track life is maximized. The trajectories that the two targets travel along are illustrated in Fig. 8, where example outputs of the JPDA and SJPDA filters are included. The angle φ (cf. Fig. 5) is $\pi/3$, and the targets travel at a maximum 10 laps around the circuits, which correspond to 1010 seconds, or almost 17 minutes. From the table we see that the SJPDA filter has much longer average track life than the JPDA filter, due to its avoidance of track coalescence and better description of where there are targets. The lower track-loss probabilities of the SJPDA filter are evident for all four detection probabilities, but the improvement compared to the JPDA filter is better the higher the detection probability.

Table 5: Average track lives and average track loss probabilities for the JPDA and SJPDA algorithms, for different detection probabilities, P_D.

		JPDA	SJPDA	Improvement SJPDA vs. JPDA
$P_D = 1$	Length	97s	993s	924%
	Loss	100%	3%	97%
$P_D = 0.95$	Length	128s	734s	473%
	Loss	100%	53%	47%
$P_D = 0.9$	Length	152s	456s	200%
	Loss	100%	88%	12%
$P_D = 0.85$	Length	142s	251s	77%
	Loss	100%	100%	0%

8 Conclusions

In this article we have shown how traditional tracking algorithms can be improved when target identity is not of interest. The presented approach uses the relation between the density of ordered (labeled) targets, and the density of unordered (the set of) targets. More specifically, there is an infinite number of ordered densities which correspond to the same unordered density, and by switching between those densities we can obtain densities which have better characteristics. In order to reflect the fact that target identity is irrelevant, we use the Mean Optimal Subpattern Assignment (MOSPA) metric instead of the Mean Squared Error (MSE).

In the first part of the article, we discuss how the new problem of minimizing MOSPA differs from the classic tracking problem, and show that there is substantial room for improvements of the traditional algorithms when evaluated according to MOSPA.

In the second part of the article we study the Joint Probabilistic Data Association (JPDA) filter and how it can be adjusted to the considered problem. First, we describe how optimal Gaussian approximations in Kullback-Leibler sense can be found. From this, we develop a new filter called the Kullback-Leibler Set JPDA (KLSJPDA) filter. Since the filter is computationally intensive, we propose another approach which operates on a goal function which leads to both good Gaussian approximations and low MOSPA. The minimization of that goal function can be formulated as a continuous optimization problem. From this, we develop another adjustment of the JPDA filter called the Set JPDA (SJPDA) filter.

In the final part of the paper, the SJPDA and KLSJPDA filters are evaluated on two simulation examples and compared to the JPDA and the Gaussian-mixture Cardinalized Probability Hypothesis Density (GM-CPHD) filters. The results show that the SJPDA and KLSJPDA filters have similar performance and that they perform substantially better than JPDA in terms of MOSPA. However, due to its lower complexity, we prefer the SJPDA filter. The results also show that the SJPDA filter has a much longer average track length than JPDA for the considered scenario. Further, it is seen that the SJPDA filter performs almost as good as the more complex GM-CPHD filter, and that it even performs slightly better for a unity detection probability.

References

[1] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, no. 5, pp. 451–460, September 1975.

[2] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE Journal of Oceanic Engineering*, vol. 8, no. 3, pp. 173–183, July 1983.

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

[4] S. Deb, M. Yeddanapudi, K. Pattipati, and Y. Bar-Shalom, "A generalized s-d assignment algorithm for multisensor-multitarget state estima-

tion," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 2, pp. 523–538, April 1997.

[5] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood, MA: Artech House, 1999.

[6] S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, January 2004.

[7] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, August 2008.

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

[9] ——, "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, October 2003.

[10] 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, November 2006.

[11] R. Mahler, "PHD filters of higher order in target number," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 4, pp. 1523–1543, October 2007.

[12] 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, July 2007.

[13] M. Guerriero, L. Svensson, D. Svensson, and P. Willett, "Shooting two birds with two bullets: How to find minimum mean OSPA estimates," in *Proceedings of the 13th International Conference on Information Fusion*, 2010.

[14] O. Drummond and B. Fridling, "Ambiguities in evaluating performance of multiple target tracking algorithms," in *Signal and Data Processing of Small Targets*, ser. Proc. SPIE, vol. 1698, 1992, pp. 326–337.

[15] J. Hoffman and R. Mahler, "Multitarget miss distance via optimal assignment," *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol. 34, no. 3, pp. 327–336, May 2004.

[16] S. Kullback, *Information Theory and Statistics*. NY: Wiley, 1959.

[17] A. Runnalls, "Kullback-Leibler approach to Gaussian mixture reduction," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 3, pp. 989–999, July 2007.

[18] L. Svensson, D. Svensson, M. Guerriero, and P. Willett, "Set JPDA filter for multi-target tracking," Department of Signals and Systems, Chalmers University of Technology, Tech. Rep. R012/2010, 2010.

[19] L. Svensson, D. Svensson, and P. Willett, "Set JPDA algorithm for tracking unordered sets of targets," in *Proceedings of the 12th International Conference on Information Fusion*, 2009.

[20] H. Blom and E. Bloem, "Probabilistic data association avoiding track coalescence," *IEEE Transactions on Automatic Control*, vol. 45, no. 2, pp. 247–259, February 2000.

[21] D. Crouse, Y. Bar-Shalom, P. Willett, and L. Svensson, "The JPDAF in practical systems: Computation and snake oil," in *Signal and Data Processing of Small Targets*, ser. Proc. SPIE, 2010.

Paper III

Performance Evaluation of MHT and CPHD in a Ground Target Tracking Scenario

D. Svensson, J. Wintenby and L. Svensson

In *Proceedings of the 12th International Conference on Information Fusion*, Seattle, USA, 2009.

Performance Evaluation of MHT and CPHD in a Ground Target Tracking Scenario

Daniel Svensson
Department of Signals and Systems
Chalmers University of Technology
Göteborg, Sweden
daniel.svensson@chalmers.se

Johannes Wintenby
Saab Microwave Systems, Saab AB
Göteborg, Sweden
johannes.wintenby@saabgroup.com

Lennart Svensson
Department of Signals and Systems
Chalmers University of Technology
Göteborg, Sweden
lennart.svensson@chalmers.se

Abstract

Performance evaluations of multi-target tracking algorithms are often limited to consider comparisons within the same algorithm family. In this paper, we evaluate two conceptually different multi-target tracking algorithms, namely a multiple-hypothesis tracking (MHT) algorithm and the Gaussian mixture cardinalized probability hypothesis density (GM-CPHD) filter. As a reference, we also compare the results to a conventional single-hypothesis tracking algorithm. The performance is assessed using the root-mean square error of the estimated number of targets, and the recently published Optimal Subpattern Assignment (OSPA) measure. The scenario under consideration is tracking of nine closely spaced ground targets, using simulated measurements from an airborne radar. The results indicate that the scenario is challenging, since the conventional algorithm does not perform nearly as well as MHT and GM-CPHD. By observing the estimation of the number of targets, as well as of the target states, we draw conclusions regarding the behavior of MHT and GM-CPHD. The main differences between the algorithms are pointed out and discussed. For example, GM-CPHD is more responsive to changes in the number of

targets, whereas MHT is less responsive, but produces a more stable output.

Keywords: Multi-target tracking, MHT, CPHD, performance evaluation, OSPA, ground target tracking.

1 Introduction

This paper is concerned with performance evaluation and comparison between three different multi-target tracking algorithms, applied to a ground target tracking scenario with closely spaced targets. The reason for performing the evaluation is that comparisons between multi-target tracking algorithms are few in the literature. Especially, there is a lack of comparisons between the Cardinalized Probability Hypothesis Density (CPHD) filter and traditional tracking algorithms. In this paper we evaluate three different types of algorithms. The first algorithm in the evaluation is a conventional Single-Hypothesis Tracking (SHT) algorithm, which uses the extended Kalman filter (EKF) [1] for filtering, Global Nearest Neighbor (GNN) [2] using the Auction algorithm [3] for data association, and track score for management of tracks [2]. The SHT algorithm is used as a reference. The main focus is on comparing a track-oriented Multiple Hypothesis Tracking (MHT) algorithm [2, 4, 5] and the recently developed Gaussian Mixture Cardinalized Probability Hypothesis Density (GM-CPHD) filter [6–9]. The reason for evaluating these two algorithms is that they are generally accepted as the high-end alternatives of their respective class of algorithms. To the best of our knowledge, no evaluation study of GM-CPHD and MHT has been published before. In the paper, we use the terms GM-CPHD and CPHD interchangeably, where both refer to GM-CPHD.

Standard SHT algorithms find and propagate only the instantaneously best data association hypothesis over time. If the association is ambiguous, it is not at all certain that this hypothesis corresponds to the true association. The family of MHT algorithms on the other hand, form and propagate a set of alternative association hypotheses. When more data is received, the least likely hypotheses can be discarded, while the true hypothesis is hopefully retained and presented to the user. The original formulation of MHT is given in [4]. In the performance evaluation of this paper, a track-oriented MHT [5,10] is used, for which the global hypotheses are reformed in each time step from the updated tracks. Efficient implementations of track-oriented MHT are described in [5, 11, 12].

The CPHD algorithm is a random finite set (RFS) approach to multi-target tracking, which has received a lot of attention in the recent years [6, 8,

9,13]. The algorithm treats the number of targets, and their respective states, as random variables. The CPHD filter recursively propagates an intensity function, defined on the single-target state space. The intensity function has the property that the integral of the function over a volume, gives the expected number of targets within that volume [14]. In the Gaussian mixture version of the algorithm, the intensity function is approximated with a sum of weighted Gaussian probability density functions (pdfs). In conjunction with the intensity function, the CPHD algorithm also propagates a so-called cardinality mass function, which gives the distribution of the total number of targets in the scene. The mass function is of arbitrary shape, which makes the CPHD filter different from its predecessor—the probability hypothesis density (PHD) filter [14, 15]. For PHD, Gaussian Mixture PHD (GM-PHD) [16] and MHT, there have been a few performance evaluations, e.g., in [17] and [18].

To assess the performance of the tracking algorithms, we use two different measures. The first is the root-mean square error (RMSE) of the cardinality estimates. The RMSE of cardinality gives a measure of how well the algorithms estimate the number of targets. In conjunction to the measure, we also plot the Monte Carlo average of the cardinality estimates over time for each filter, to see the difference between how the filters respond to changes in the number of targets. The second measure that we use is the recently developed Optimal Subpattern Assignment (OSPA) measure [19, 20], which is based on the Wasserstein metric for multi-target tracking [21]. The OSPA measure tries to capture the quality of both the cardinality and the state estimation.

The scenario considered in the evaluation concerns tracking of nine targets on the ground, using simulated measurements from an airborne radar. The targets are closely spaced, and their trajectories cross. In the middle part of the scenario, a small mountain obscures up to two targets from the sensor. During that period of time, not all targets are detectable. The scenario has been designed to be challenging for any multi-target tracking algorithm.

In the performance evaluation, some observations are made. First, the standard SHT algorithm performs worse than the more advanced MHT and GM-CPHD algorithms, which indicates that the considered scenario is indeed challenging. Secondly, the OSPA overall performance is approximately equal for the MHT and CPHD approaches, with a slight advantage for CPHD. We note three differences between the behavior of MHT and GM-CPHD:

1) The output of MHT is more stable than the CPHD output, due to the slower response of MHT to new and vanishing targets, which has to do with how tracks are initiated and deleted. The standard deviation of

the estimate of the number of targets is overall lower for MHT than for CPHD.

2) The CPHD filter has a faster response to vanishing targets, than MHT. This is due to the CPHD modeling of vanishing targets by a death model. In MHT, there is no such model; instead, target tracks are removed when their quality is deemed too low. Whether the fast response is beneficial or not, is a matter of application. If a target is just temporarily undetectable, a fast response has the drawback of lost track continuity.

3) The response to increasing numbers of targets is faster in CPHD, than in MHT. The reason is that the probability calculation in the considered MHT approach is started only for tracks with 3 detections or more. There is hence normally a delayed presentation of new targets of three time steps.

2 Scenario and Problem Formulation

The scenario considered in this paper is the tracking of nine ground targets, using simulated measurements given by an airborne radar. The tracking is challenging due to several reasons:

1) The targets are closely spaced, and their trajectories cross at numerous times.

2) Due to terrain masking, all targets are not visible to the sensor during the entire scenario.

3) Since the sensor is far off in the beginning of the scenario, the SNR is low, which gives an initially low detection probability P_D.

4) On top of target-generated detections, the sensor also reports spurious detections due to clutter.

In Figure 1, we plot the trajectories of the nine targets. As we see, the targets are close together during a large part of the scenario, and their trajectories cross.

The state vector $\mathbf{x}_k$ of a target has six components

$$\mathbf{x}_k = [x_k \; y_k \; z_k \; \dot{x}_k \; \dot{y}_k \; \dot{z}_k]^T, \tag{1}$$

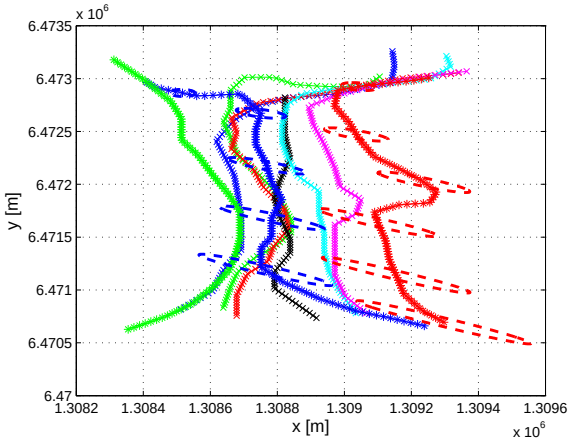

Figure 1: Trajectories of the nine targets in the considered ground target tracking scenario. The distances between grid points are 200 m and 500 m in the x and y dimensions, respectively. The dashed ellipses illustrate the sensor accuracy for a few instances of time. As the sensor approaches the group of targets, the accuracy improves.

namely position and speed in three dimensions. The motion model used is a nearly constant-velocity model

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{v}_{k-1}, \tag{2}$$

with system matrix

$$\mathbf{F}_{k-1} = \begin{bmatrix} \mathbf{I}_{3\times 3} & T_0\mathbf{I}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{I}_{3\times 3} \end{bmatrix}. \tag{3}$$

In (3), $\mathbf{I}_{m\times m}$ is an $m \times m$ identity matrix, and $\mathbf{0}_{m\times n}$ a $m \times n$ matrix of zeroes. The process noise $\mathbf{v}_{k-1}$ is zero-mean Gaussian with covariance matrix $\mathbf{Q}_{k-1} = \mathbf{Q}$, where

$$\mathbf{Q} = \begin{bmatrix} q_0 T_0^3/3\mathbf{I}_{2\times 2} & \mathbf{0}_{2\times 1} & q_0 T_0^2/2\mathbf{I}_{2\times 2} & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & q_z T_0^3/3 & \mathbf{0}_{1\times 2} & q_z T_0^2/2 \\ q_0 T_0^2/2\mathbf{I}_{2\times 2} & \mathbf{0}_{2\times 1} & q_0 T_0 \mathbf{I}_{2\times 2} & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & q_z T_0^2/2 & \mathbf{0}_{1\times 2} & q_z T_0 \end{bmatrix}. \tag{4}$$

In (4), $q_0 = 3$ and $q_z = 0.01$. The uncertainty is lower in the z dimension, since we know that the targets are moving on the ground. In the scenario, all nine targets move with a constant speed of 5 m/s, and they move both on roads and in the terrain. The targets have randomly varying radar cross

sections. The target trajectories are not generated according to the motion model. Instead, they have been manually created using a map-based tool which includes 3D elevation and terrain type databases.

Measurements are provided by a radar sensor, which measures range and angle, according to

$$\mathbf{z}_k = [r_k \ \varphi_k \ \theta_k]^T , \tag{5}$$

where r_k is range, φ_k azimuth and θ_k elevation. The time between measurements is $T_0 = 3$ s. The measurement model

$$\mathbf{z}_k = h_k(\mathbf{x}_k) + \mathbf{w}_k \tag{6}$$

is thus non-linear, with the measurement function

$$h_k(\mathbf{x}_k) = \begin{bmatrix} \sqrt{x_k^2 + y_k^2 + z_k^2} \\ \arctan(y_k/x_k) \\ \arctan\left(z_k/\sqrt{x_k^2 + y_k^2}\right) \end{bmatrix} . \tag{7}$$

The measurement noise $\mathbf{w}_k$ is independent of the process noise $\mathbf{v}_{k-1}$, and is zero-mean Gaussian with covariance matrix

$$\mathbf{R}_k = \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_\varphi^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix} . \tag{8}$$

The measurement accuracies (standard deviations) are $\sigma_r = 5$ m, $\sigma_\varphi = 0.003$ rad and $\sigma_\theta = 0.003$ rad. The accuracy in range is much better than the angular accuracy, as illustrated in Figure 1. The platform carrying the sensor flies at an altitude of 5 km, and its range to the targets varies from approximately 60 km to 6 km. In Figure 2, the movement of the sensor platform over time is shown. The figure also includes the positions of the nine targets over time, as well as the mountain which obscures up to two targets from the sensor. The target visibility from the platform position is calculated using the terrain and elevation databases.

The clutter measurements are assumed independent from scan to scan, and spatially uniform in the x-y plane (cf. Figure 2). The density of clutter is $0.1/\text{km}^2$, which is reasonable in the considered application with large surveillance volume. Sensitivity analysis, with varying clutter densities, is left for future work.

Since the measurement model is non-linear, we need a non-linear filtering algorithm. In this paper, we use the Extended Kalman Filter (EKF) [1].

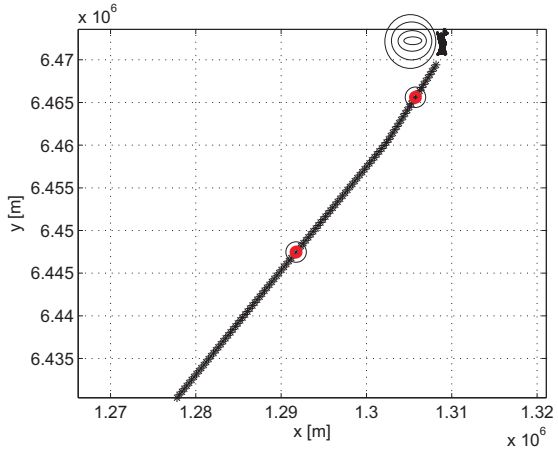

Figure 2: Illustration of the ground target tracking scenario. The line starting from the lower left corner represents the movement of the sensor platform over time. The black area in the upper right corner represent the trajectories of the nine targets. To the left of the group of targets, a mountain is located, which is marked with altitude lines. The two thick circles on the sensor platform trajectory represent the start and end position of the sensor, for which $1-2$ of the targets are obscured by the mountain.

The problem under consideration is the tracking of multiple targets. We are hence interested in a recursive estimation of both the number of targets and their respective state vectors. At each time step k, we thus seek an estimate $\hat{N}_{k|k}$ of the number of targets, given the sequence of measurements, described by the ordered set $\mathbf{Z}_{1:k} = \{\mathbf{z}_1, \ldots, \mathbf{z}_k\}$. For the $\hat{N}_{k|k}$ targets, we want to express the pdfs $p\left(\mathbf{x}_k^{(i)} | \mathbf{Z}_{1:k}\right)$ for $i = 1, \ldots, \hat{N}_{k|k}$.

2.1 Performance Measures

In order to assess the multi-target tracking algorithms, we need one or more performance measures. In this paper, we use two different measures of performance. The first one is a measure of the *cardinality* estimation, i.e., how well the algorithms estimate the number of targets. The name comes from set theory, where the cardinality of a set is the number of elements in the set. The cardinality measure used is the root-mean square error (RMSE)

$$e_{\text{card}}^{\text{RMSE}}(k) = \sqrt{\frac{1}{M} \sum_{l=1}^{M} \left(\hat{N}_{k|k}^{l} - N_k^{\text{true}}\right)^2}, \tag{9}$$

where M is the number of Monte Carlo simulations, $\hat{N}_{k|k}^{l}$ is the estimate of cardinality at time k for the l^{th} iteration, and N_k^{true} is the true number of targets at time index k. In the scenario considered in this paper, the true number of targets could either mean the actual number of targets in the scenario, or the number of visible targets at time k, depending on the desired behavior of the tracking algorithm. We will primarily consider the number of visible targets in the evaluation.

On top of the cardinality performance, we also need a measure on how well the algorithms estimate the target states. Since the number of targets is not known, it is not straight-forward to devise such a measure. We use the recently published Optimal Subpattern Assignment (OSPA) measure [19,20]. The OSPA measure tries to capture the quality of both the cardinality and the state estimation. Let $\mathbf{Y}$ be the set of true target states and $\mathbf{X}$ be the set of target estimates, with cardinalities n and m, respectively. The OSPA measure $\bar{d}_p^{(c)}$ is then defined as

$$\begin{aligned}&\bar{d}_p^{(c)}(\mathbf{X},\mathbf{Y})\\&\quad=\left(\frac{1}{n}\left(\min_{\pi\in\Pi_n}\sum_{i=1}^{m}d^{(c)}(\mathbf{x}_i,\mathbf{y}_{\pi(i)})^p+c^p(n-m)\right)\right)^{1/p},\end{aligned}\tag{10}$$

if $m \leq n$ and $\bar{d}_p^{(c)}(\mathbf{X},\mathbf{Y}) = \bar{d}_p^{(c)}(\mathbf{Y},\mathbf{X})$ otherwise. Here, $d^{(c)}(\mathbf{x},\mathbf{y}) \triangleq \min(c, d(\mathbf{x},\mathbf{y}))$ is the distance d between $\mathbf{x}$ and $\mathbf{y}$, cut-off at c. Further, Π_n is the set of all possible permutations of $\mathbf{Y}$. In this article, we let d be the Euclidean distance. In practice, the measure performs an optimal assignment of target estimates to true target states. A performance evaluation of PHD and MHT using the Wasserstein distance is found in [17].

3 Tracking Algorithms

In this paper, we compare three multi-target tracking algorithms. The first one is a multiple hypothesis tracking (MHT) algorithm, with probability-based pruning combined with N-scan pruning [2]. The second one is the recently published Gaussian Mixture Cardinalized Probability Hypothesis Density (GM-CPHD) filter [6–9]. The third one is a conventional single-hypothesis tracking (SHT) algorithm.

3.1 MHT

The MHT algorithm in the evaluation is a track-oriented algorithm derived for general radar target tracking. In the algorithm, hypothesis control is made

purely with pruning, which utilizes a combination of low probability pruning of tracks and N-scan pruning. Probabilities are computed approximately by generating the K-best global hypotheses in local neighborhoods of each target, using methods from [22]. In a track-oriented fashion, the hypothesis generation is carried out every new data frame, given track incompatibility tables. The probability calculation was designed such that, at a maximum, the computation time is roughly a millisecond per target in arbitrary scenarios, and given 20 track hypotheses per target. In practice, the time used for probability calculation is small compared to that of the prediction, gating and hypothesis spawning steps of the algorithm. In standard radar scenarios, the side-effects of the approximations can be ignored. However, in the scenario considered in this paper, the performance may suffer somewhat due to the high target density, which results in many incompatibilities between tracks, and consequently a challenging data association problem. In the performance evaluations based on Monte Carlo simulation, see Figure 3 to 6, the downgrade due to the approximation is estimated to be less than, say, 10%.

The initiation step in the algorithm was designed for high clutter density scenarios, and the probability calculation is started only for tracks with 3 detections or more. A consequence of this is a delayed presentation of new targets in comparison to the CPHD filter, see Section 4 for a discussion. Probability of target existence is also calculated for every potential target that has entered the probability calculation, and only targets with high probability of existence are presented.

In contrast to the CPHD filter, the MHT-algorithm lacks a model for target death, or the event that a target becomes invisible to the sensor. In the scenario, two targets disappear. To MHT, these targets still exist and the tracks are removed only after a while when their quality is considered poor. The CPHD estimates the number of visible targets, and therefore reacts faster to changes in the number of visible targets. If this is good or bad is probably a matter of choice. The issue is discussed further in Section 4.

A major difference between MHT and CPHD is that MHT produces track-valued estimates, whereas the standard CPHD filter produces state-valued estimates, where the trajectories are restored in retrospect. This might have an effect on track continuity, which is often essential to the user, i.e., the ability to tell how a target has moved from that it became visible until it leaves the scenario. In the quantitative performance comparisons herein, track continuity is not evaluated.

3.2 CPHD

The CPHD algorithm is a philosophically different approach to multi-target tracking, compared to, e.g., MHT. It models targets as a random finite set (RFS). The tracking problem is to calculate the conditional density of the target RFS $\mathbf{X}_k$, conditioned on the set $\mathbf{Z}_{1:k}$ of measurements up to time k. To that tracking problem, the CPHD algorithm provides an approximate solution. The algorithm recursively estimates the so-called probability hypothesis density (*phd*), or intensity function, which is the first-order statistical moment of the RFS $\mathbf{X}_k$ [23]. The *phd* describes the intensity of targets in the target state-space, and it has the property that the integral of the *phd* over a region in the state-space, gives the expected number of targets within that region. That is, if $v_{k|k}(\mathbf{x}_k)$ is the posterior intensity function at time k,

$$\int_{\mathcal{S}} v_{k|k}(\mathbf{x}_k)\, d\mathbf{x}_k = \mathbb{E}\{|\mathbf{X}_k \cap \mathcal{S}|\} \tag{11}$$

describes the expected number of targets in the region $\mathcal{S}$ of the single-target state-space. In (11), $|\cdot|$ is the cardinality of a set, and $\mathbf{X}_k$ is the target posterior RFS at time k.

To be able to provide a good estimate on the cardinality of $\mathbf{X}_k$, the CPHD filter also propagates the full cardinality distribution $\Pr\{n_k|\mathbf{Z}_{1:k}\}$ in time. From the cardinality distribution, a MAP estimate of the number of targets is given by

$$N_{k|k}^{\text{MAP}} = \arg\max_{n_k} \Pr\{n_k|\mathbf{Z}_{1:k}\}. \tag{12}$$

The CPHD algorithm rests on the following assumptions:

- Each target evolves and generates measurements independently of the others.
- Clutter is a cluster RFS, defined below, and independent of the target-generated RFS.
- Predicted and posterior target RFSs are approximated as cluster RFSs.
- The birth RFS is a cluster RFS, and it is independent of the surviving target RFS.

A cluster RFS is an RFS with independent and identically distributed elements, and where the number of elements in the set is described by an

arbitrary cardinality distribution. CPHD thus assumes that targets are independent and identically distributed. If the cardinality distribution is Poisson, the cluster RFS is called a Poisson RFS. It is common to model clutter as a Poisson RFS, and it is also the approach of this paper.

New targets in the scene are modeled by a birth process. Target disappearance from the scene is modeled by a death process, where targets vanish with a certain probability. The CPHD filter has so far only been implemented using a Gaussian mixture approach, presented in [7, 8], where the intensity function is approximated by a weighted sum of Gaussian components. It is also the implementation used in this paper. In GM-CPHD, a common choice is to represent births with a large Gaussian component in the middle of the field of view, and with a small weight corresponding to the probability of the appearance of a new target. In the death model of GM-CPHD, each mixture component survives between time steps with a survival probability, P_S. The filter operates in real time on the considered ground target tracking scenario.

4 Results

In this section, we present and discuss the results obtained when applying the SHT, MHT and CPHD algorithms on a ground target tracking scenario with 9 targets. The parameter setup for the tracking algorithms is as follows. For MHT, a constant detection probability $P_d = 0.8$ is assumed throughout the scenario. The algorithm is not sensitive to this selection. There are further parameters, such as the intensity of new targets and the clutter density, and these parameters have been tuned for the scenario. For CPHD, $J_{\max} = 80$ is the maximum number of mixture components after pruning and merging. Further, the detection probability $P_d = 0.85$ and the survival probability $P_S = 0.91$. Different detection and survival probabilities have been tested, and the selection is of some importance. For merging, $U = 4$ is the statistical distance threshold, and for the pruning, mixture components with weight less than 0.001 are removed. The birth model is represented by a Gaussian component centered in the scenario, with a weight $w_{\text{birth}} = 0.001$, which also is the probability of a new target emerging. The birth component has a covariance matrix, whose 1-σ ellipse covers the field of view of the sensor, in Cartesian coordinates. For SHT, track score parameters, such as false alarm rate and probability of true track deletion, have been tuned for the scenario. The SHT is, however, not very sensitive to the setup used.

The results are given as three quantities. The first one is the average cardinality estimate of the three algorithms, which is shown in Figures 3 and 4. The second quantity is the average cardinality error compared to the

true visible number of targets, given in Figure 5. Finally, the average OSPA performance measure of MHT, CPHD and SHT is presented in Figure 6. For OSPA, $p = 2$ and $c = 300$ m. The results are based on 100 Monte Carlo simulations.

We start by discussing the cardinality estimation of the tracking algorithms. In Figure 3, we see the average cardinality estimates over time. The result from time step 40 and onward is given in Figure 4. For CPHD, the MAP estimate in (12) is used. For SHT and MHT, the cardinality is the number of presented target tracks. A track is presented if its probability of existence supersedes a preset threshold. By observing Figures 3 and 4, we can draw some conclusions. First, the MHT algorithm is slower in adapting to an increase in cardinality. Initially, the number of targets is 0, and the presented number of targets is not equal to 9 until after some 50 seconds, which corresponds to approximately 15 time steps. The CPHD algorithm, on the other hand, is fast in adapting to changes in cardinality. In the early part of the scenario, CPHD quickly estimates the number of targets to 7. The estimate is then slowly increasing to 9. In the end part of the scenario, we note that the change in the cardinality estimate is almost immediate. We also notice that MHT has a fairly fast response to the changes in number of visible targets at time instances 250 and 300 seconds of the scenario, at which the obscured targets are again visible.

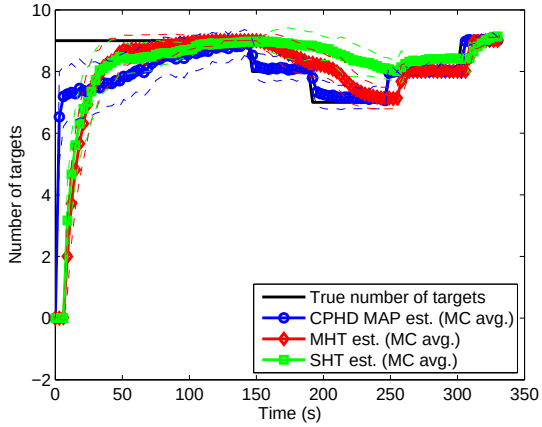

Figure 3: True number of targets, and average cardinality estimates of SHT, MHT and CPHD for a ground target tracking scenario with 9 targets. The dashed lines represent the standard deviation of the estimates. For a period of time, $1-2$ targets are obscured by a mountain and are not detectable by the airborne sensor.

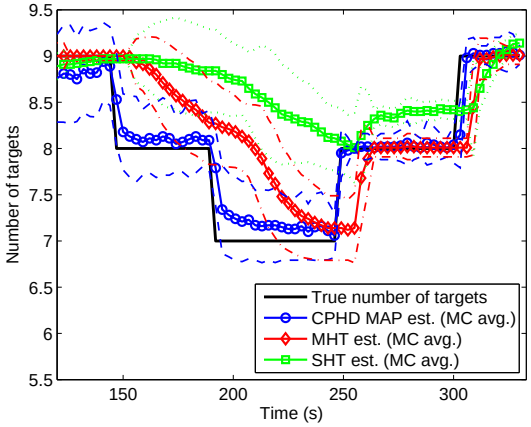

Figure 4: True number of targets, and average cardinality estimates of SHT, MHT and CPHD for a ground target tracking scenario with 9 targets. The results from time step 40 and forward is presented. The dashed, dash-dotted and dotted lines represent the standard deviation of the estimates of the respective algorithm. The number of visible targets is reduced when a mountain obscures $1-2$ targets from the airborne sensor.

The second thing we observe in Figures 3 and 4 is that the MHT algorithm provides a stable estimate of cardinality, once the cardinality estimate is equal to the true cardinality. This has to do with the probability calculation in the algorithm, where a track needs to have been associated with three detections, before the probability calculation is initiated. This initiation procedure saves computational power, and it also provides a robust algorithm, which does not start up low-quality tracks. On the other hand, it introduces a lag of three time steps, which is noticed in Figure 4 at times following cardinality changes. It is also seen that the standard deviations of the MHT cardinality estimate is lower than for CPHD, which is a result of the robust procedure.

A third observation from Figures 3 and 4 is the behavior of the MHT and CPHD algorithms, when two of the targets become obscured by a mountain, i.e., when the true visible number of targets decrease. The CPHD algorithm is responsive to this cardinality decrease, while the MHT algorithm is slower to respond. The reason for this is due to the death process of the CPHD algorithm, which models the possibility of target disappearance. The MHT algorithm, on the other hand, does not have such a death model. Instead, targets are removed if their likelihood have decreased below some threshold. The MHT thus becomes slower in reducing the cardinality, but on the other hand keeps the tracking continuity over shorter periods of target invisibility.

No method is uniformly better than the other, but depends on the application and what behavior that is preferred. A positive aspect of a slower response to a cardinality decrease, is if it is short-term, i.e., if the target is only intermittently undetectable. A less responsive algorithm then allows for track continuity, by not removing the target track. Another approach would be to introduce a death model in MHT, to allow for a fast response to cardinality reductions, and then to have a track-connection algorithm, which connects target tracks that appear to stem from the same target. Development of such a procedure is left for future work.

In Figure 5, we plot the cardinality error over time, where the error is in relation to the true number of visible targets. We notice that the SHT and MHT algorithms are slow at the startup of the scenario, but that they after 25 seconds give more accurate cardinality estimates than CPHD. Notably, between time 100 and 145 seconds of the scenario, MHT provides a stable and accurate estimate of the number of targets. Then, at each cardinality change, there is a period of higher RMSE for MHT, due to the slow response. This is a bias error of the cardinality estimate. For CPHD, after the filter has tuned in on the correct number of targets, the filter provides cardinality estimates that are less fluctuating, compared to the number of visible targets.

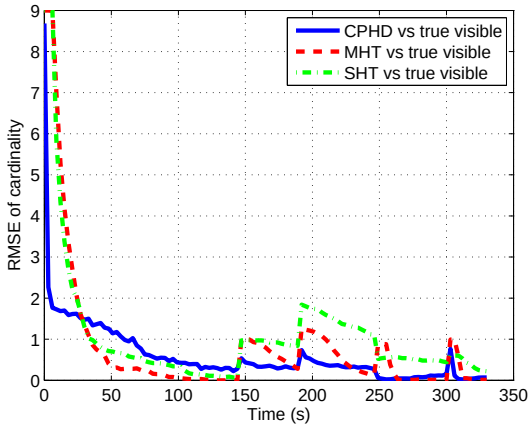

Figure 5: RMSE of cardinality estimation error of CPHD, MHT and SHT for a ground target tracking scenario with $7-9$ visible targets.

The overall performance of the filters, including both the cardinality estimation and the state estimation, is evaluated by the OSPA measure. A low value indicates good performance. In Figure 6, the average OSPA distance is plotted over time. Some conclusions can be drawn. First, we observe that the measure captures the cardinality estimation performance, since the

OSPA measure increases when the cardinality estimate becomes less accurate as compared to the true number. From the point of view of the OSPA measure, it is worse to produce an incorrect cardinality estimate when the measurement accuracy is high, compared to when it is low. This is noticed by studying the OSPA measure at times 50 and 250. At time 50, the MHT algorithm has a cardinality RMSE, which is approximately one unit less than that of CPHD. In contrast, at time 250, the CPHD RMSE of cardinality is approximately one unit less than is the MHT RMSE. The OSPA differences between the filters are, however, not of the same order at these two time instants. The only difference in the scenario between the early and the late part is that the measurement accuracy is higher in the end, due to the sensor being closer to the targets.

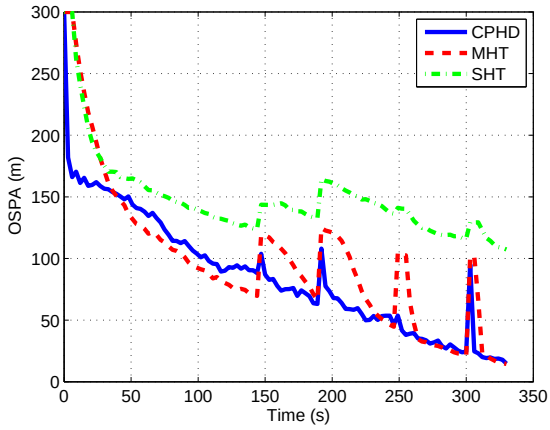

Figure 6: Performance of CPHD, MHT and SHT for a ground target tracking scenario, with $7-9$ visible targets. The measure of performance is the OSPA measure, with $c = 300$ m and $p = 2$.

Another property that can be noticed in Figure 6, is that the MHT performance is more stable than the CPHD output, which is concluded by studying the variation of the measure over time. This is due to the track initiation and deletion procedures of MHT. When the cardinality estimates of the CPHD and MHT filters are equal, their OSPA measures are also approximately equal, so the output states are of the same quality for the respective filters. However, if the merging threshold of CPHD is reduced, e.g., to $U = 1$, the OSPA for CPHD is slightly higher than for MHT, at times when the cardinality estimates are equal. CPHD thus benefits from merging mixture components which are close together. In Figure 6, we also see that the width of the peaks in the OSPA for MHT are wider when the cardinality is de-

creased, than when it is increased. MHT is thus quite responsive to new targets appearing (or re-appearing), but slower at responding to vanishing targets. Introducing a death model for targets would increase the responsiveness of the algorithm. Also, it is not always the case that a fast response to vanishing targets is requested by the user; instead, having slower response, with the gain of increased track continuity could be more attractive.

In the parameter setup of CPHD, the parameters have been selected to yield a fast response to both increasing and decreasing number of targets. The parameters could instead have been adopted to yield a slower response to a decreasing number of targets, e.g., by lowering the survival probability P_S, which would give a behavior more similar to MHT. Which behavior that is best is a matter of application.

5 Conclusions

In this paper, we have evaluated the multiple hypothesis tracking (MHT) and the cardinalized probability hypothesis density (CPHD) filters in a ground target tracking scenario with an airborne radar sensor. As reference, we have also evaluated a conventional single-hypothesis tracking (SHT) filter for the same scenario. To assess the performance, two measures have been used: the root-mean square error of the cardinality estimates, and the OSPA measure introduced in [19, 20]. The number of targets in the scenario is 9. The targets are close together, and their trajectories cross at numerous times. For a period of time, up to two of the targets are not visible to the radar sensor, due to obscuration by a mountain. It is hence a challenging scenario.

In the evaluation, we conclude that the MHT and CPHD filters perform approximately equally well overall, but that the filters have two properties which make them different. The first one regards how the number of targets are estimated. In CPHD, a maximum a posteriori estimate is derived from the cardinality distribution. In MHT, on the other hand, the estimate is equal to the number of presented targets, where a target is presented if its probability of existence supersedes a threshold. The second property relates to how the MHT and CPHD filters respond to increasing and decreasing number of targets. The CPHD filter is more responsive to such changes than MHT. For increasing number of targets, it is due to the track initiation procedure of MHT, which requires three detections or more for a track in order to start the probability calculation. For decreasing number of targets, the difference lies in how the filters model vanishing targets. In CPHD, there is a death model that states that a target vanishes with a certain probability, which makes the filter quick at adapting to reductions in the number of

visible targets. In MHT, targets are removed of their likelihood falls below a threshold. This makes the algorithm less responsive to decreases in the number of visible targets. On the other hand, it yields a robust method, which produces more stable outputs and which can keep track continuity over short periods of target invisibility. The MHT produces estimates of the number of targets which overall have lower standard deviation than the CPHD estimates, which is a result of the slow response. Introducing a death model for MHT could be a way to make the algorithm more responsive to vanishing targets, if desired.

A third, interesting measure in the comparison would have been that of computational burden. Currently, such a comparison is not possible, since the algorithms are implemented in different frameworks, with different possibilities of optimization in the compilation. However, they both enable real-time operation.

Acknowledgement

This research has been supported by the National Aeronautic Research Program (NFFP), which is funded by VINNOVA (The Swedish Governmental Agency for Innovation Systems). The research project is a joint collaboration between the signal processing group at Chalmers University of Technology and Saab Microwave Systems, Saab AB.

References

[1] A. Jazwinski, *Stochastic Processes and Filtering Theory.* New York: Academic, 1970, ch. 8, pp. 272–281.

[2] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems.* Norwood, MA: Artech House, 1999.

[3] D. Bertsekas, "The auction algorithm: a distributed relaxation method for the assignment problem," *Annals of Operations Research*, vol. 14, no. 1–4, pp. 105–123, June 1988.

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

[5] S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, January 2004.

[6] R. Mahler, "Phd filters of higher order in target number," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 4, pp. 1523–1543, October 2007.

[7] B.-T. Vo, B.-N. Vo, and A. Cantoni, "The cardinalized probability hypothesis density filter for linear Gaussian multi-target models," in *40th Annual Conference on Information Sciences and Systems*, March 2006.

[8] ——, "Analytic implementations of the cardinalized probability hypothesis density filter," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3553–3567, July 2007.

[9] M. Ulmke, O. Erdinc, and P. Willett, "Gaussian mixture cardinalized PHD filter for ground moving target tracking," in *Proceedings of the 10th international conference on information fusion*, July 2007.

[10] T. Kurien, "Issues in the design of practical multitarget tracking algorithms," in *Multitarget-Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom, Ed. Norwood, MA: Artech House, 1990.

[11] G. Demos, "Applications of MHT to dim moving targets," in *Proceedings of the SPIE - The International Society for Optical Engineering*, vol. 1305, April 1990, pp. 297–309.

[12] E. Fortunato, W. Kreamer, S. Mori, C.-Y. Chong, and G. Castanon, "Generalized Murty's algorithm with application to multiple hypothesis tracking," in *Proceedings of the 10th international conference on information fusion*, July 2007.

[13] M. Ulmke, D. Fränken, and M. Schmidt, "Missed detection problems in the cardinalized probability hypothesis density filter," in *Proceedings of the 11th International Conference on Information Fusion*, June–July 2008.

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

[15] ——, "Multitarget bayes filtering via first-order multitarget moments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, October 2003.

[16] 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, November 2006.

[17] K. Panta, B. Vo, A. Doucet, and S. Singh, "Probability hypothesis density filter versus multiple hypothesis testing," in *Signal Processing, Sensor Fusion and Target Recognition XIII*, ser. Proc. SPIE, vol. 5429, 2004.

[18] D. Clark, K. Panta, and B.-N. Vo, "The GM-PHD filter multiple target tracker," in *Proceedings of the 9th International Conference on Information Fusion*, 2006.

[19] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "On performance evaluation of multi-object filters," in *Proceedings of the 11th International Conference on Information Fusion*, June-July 2008.

[20] ——, "A consistent metric for performance evaluation of multi-object filters," *IEEE Transaction on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, August 2008.

[21] J. Hoffman and R. Mahler, "Multitarget miss distance via optimal assignment," *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol. 34, no. 3, pp. 327–336, May 2004.

[22] J. Wintenby, "Probability evaluation in MHT with a product set representation of hypotheses," in *Proceedings of the 9th International Conference on Information Fusion*, July 2006.

[23] R. Mahler, "A theoretical foundation for the stein-winter probability hypothesis density (phd) multitarget tracking approach," in *Proceedings of the 2000 MSS National Symposium on Sensor and Data Fusion*, vol. I (unclassified), San Antonio, TX, USA, June 2000, pp. 99–117.

Paper IV

Multitarget Sensor Resolution Model and Joint Probabilistic Data Association

D. Svensson, M. Ulmke and L. Danielsson

Submitted to
IEEE Transactions on Aerospace and Electronic Systems.

Preliminary results have been published as:

D. Svensson, M. Ulmke and L. Danielsson,
"Multitarget sensor resolution model for arbitrary target numbers",
in *Signal and Data Processing of Small Targets*,
ser. Proc of SPIE, vol. 7698, Orlando, USA, 2010.

D. Svensson, M. Ulmke and L. Danielsson,
"Joint probabilistic data association filter
for partially unresolved target groups",
in *Proceedings of the 13th International Conference
on Information Fusion*, Edinburgh, UK, 2010.

Multitarget Sensor Resolution Model and Joint Probabilistic Data Association

Daniel Svensson*, Martin Ulmke†, and Lars Danielsson‡

Abstract

In the design of target tracking algorithms, the aspect of sensor resolution is rarely considered. Instead, it is usually assumed that all targets are always resolved, and that the only uncertainties in the data association are which targets that are detected, and which measurement each detected target gave rise to. However, in situations where the targets are closely spaced in relation to the sensor resolution, this assumption is not valid, and may lead to degraded tracking performance due to an incorrect description of the data. In this paper, we present a framework for handling sensor resolution effects for an arbitrary, but known, number of targets. We propose a complete multitarget sensor resolution model that can be incorporated into traditional Bayesian tracking filters. Further, the exact form of the posterior probability density function is derived, and two alternative ways of approximating that exact posterior density with a Joint Probabilistic Data Association (JPDA) filter are proposed. Evaluations of the resulting filters on simulated radar data show significantly increased tracking performance compared to the JPDA filter without a resolution model.

Index Terms

Target tracking, state estimation, Bayes methods, sensor models, radar detection, radar resolution

*D. Svensson is with the Department of Signals and Systems, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden, E-mail: daniel.svensson@chalmers.se.

†M. Ulmke is with the Department of Sensor Data Fusion, Fraunhofer-FKIE, Wachtberg, Germany, E-mail: martin.ulmke@fkie.fraunhofer.de.

‡L. Danielsson is with Active Safety Electronics, Volvo Car Corporation, Gothenburg, Sweden, E-mail: ldanie25@volvocars.com.

1 Introduction

Traditionally, the aspect of sensor resolution has not been considered in the design of target tracking algorithms [1]. Instead, it has been assumed that the targets are always resolved, and that the data association problem only regards the assignment of measurements to single targets. In many situations, that assumption is reasonable, but there are important cases when the resolution limitations of the sensor cannot be ignored [2]. Such cases arise when targets are temporarily closely spaced in relation to the resolution of the sensor, e.g., when tracking aircraft in formation, or in convoy tracking for ground surveillance. For such applications, ignoring the limited resolution of the sensors may lead to an incorrect interpretation of the data, which results in degraded performance, in particular due to premature deletion of tracks.

To account for resolution in a tracking algorithm, the resolution phenomena must be modeled. One way of modeling them is to express the capability of the sensor to resolve individual targets in a group by a resolution probability, and to have a model for the merged measurement that arises when the targets are unresolved. Two approaches have been proposed in the literature which follow this line of modeling. In [3], a grid-based resolution model for two targets is proposed, where the resolution probability is zero if the targets are within the same resolution cell, and one otherwise. The integration into the Joint Probabilistic Data Association (JPDA) filter [4] is also described, and in [5] the model is further extended to a Multiple Hypothesis Tracking (MHT) framework [1, 6–8]. Even though a fixed grid often is a good approximation of the signal processing procedure, the capability of a sensor to resolve targets generally depends on their positions relative to the sensor. Hence, in [9], a simple, but qualitatively correct, resolution model that take relative positions into account is proposed. Further, its incorporation into an MHT filter is also presented. However, just as the model in [3], the approach is restricted to only consider two targets. An alternative approach to the tracking of closely spaced objects is to treat them as a group. For an overview on group target tracking, see [10].

In this article, we consider the modeling of limited sensor resolution for an arbitrary, but known, number of targets. It is hence the first approach to consider resolution problems for more than two closely-spaced targets. The main contribution of the article is a framework for handling resolution limitations, which can be easily incorporated into a Bayesian tracking setting. The framework relies on a graph description of a resolution event, and on modeling the resolution probability as independent between target pairs. To complete the framework and to attain a multitarget resolution model, a model for the resolution probability for two targets and a group measurement

model for an arbitrary number of targets are needed; for example, the models in [3] or [9]. Preliminary results have previously been published in [11, 12].

The outline of the article as follows. In Section II, the problem formulation is stated. The proposal of a framework for sensor resolution modeling is given in Section III, together with the graph description of a resolution event. To complete the framework, specific measurement and resolution models for radar sensors are suggested in Section IV. In Section V, the exact calculation of the posterior pdf under both resolution and data association conflicts is described, while a Gaussian-mixture approximation is presented in Section VI. In Section VII, two alternative approaches of incorporating the resolution model into the JPDA filter are presented, and in Section VIII those approaches are evaluated on simulated radar data. In the evaluation, the tracking performance is compared to that of the JPDA filter without a resolution model. The results show improved performance for all considered setups. Finally, in Section IX, conclusions are drawn.

2 Problem formulation

The general problem considered in this article is tracking of a known number of targets under resolution limitations and unknown data associations. To this problem there are several subproblems. First, to model the probability that a group of targets is unresolved; second, to model the corresponding merged group measurement; and third, to find how these models can be incorporated into a general tracking framework.

The kinematic states of the N targets are represented by a joint state vector

$$\mathbf{x}_k = \left[\left(\mathbf{x}_k^{(1)}\right)^T \quad \left(\mathbf{x}_k^{(2)}\right)^T \quad \dots \quad \left(\mathbf{x}_k^{(N)}\right)^T \right]^T, \tag{1}$$

where $\mathbf{x}_k^{(i)}$ is the state of target i. At each time instant, t_k, a sensor produces measurements $\mathbf{Z}_k$ which contain information regarding the kinematic states $\mathbf{x}_k$. The collection of measurements from discrete time index 1 to index k is represented by

$$\mathbf{Z}^k = \left\{ \mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_k \right\}. \tag{2}$$

The goal of a tracking filter is to recursively calculate the posterior density $p\left(\mathbf{x}_k \middle| \mathbf{Z}^k\right)$, from which optimal state estimates can be derived. To be able to calculate the density, models for the sensor measurements and for the motion of the targets are necessary. Additionally, when limited sensor resolution is considered, a model that describes that phenomenon is also needed. In the

following, the considered process and measurement models are presented, while radar resolution models are proposed and discussed in Sections 3 and 4.

2.1 Process model

The process model, often called the motion model, describes the dynamics of the targets. Given the state $\mathbf{x}_{k-1}$ at time index $k-1$, the model describes the evolution of the state up to time index k. For the general case,

$$\mathbf{x}_k = f_{k-1}(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}), \tag{3}$$

where f_{k-1} is the system function describing the transition from time $k-1$ to time k, and $\mathbf{v}_{k-1}$ is a process noise that describes the uncertainties in the prediction. For Gauss-Markov systems, which we mainly consider in this article, it holds that

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{v}_k, \tag{4}$$

where $\mathbf{F}_{k-1}$ is the multi-target system matrix at time $k-1$, and $\mathbf{v}_k$ is Gaussian distributed with zero mean and covariance matrix $\mathbf{Q}_k$.

2.2 Sensor model

The sensor model describes the relation between the received measurements and the target states. The measurements received at a certain time index include both target-generated detections and spurious measurements that are due to false alarms and clutter (henceforth subsumed as clutter).

The joint measurement vector at time k is given by

$$\mathbf{Z}_k = \left[\left(\mathbf{z}_k^{(1)}\right)^T \quad \left(\mathbf{z}_k^{(2)}\right)^T \quad \dots \quad \left(\mathbf{z}_k^{(M_k)}\right)^T \right]^T. \tag{5}$$

The heritage of the data is unknown, which means that it is not known which M_c measurements are clutter and which M_t measurements are target-generated. Additionally, it is not known which target or targets gave rise to each of the true detections.

The uncertainty in the discrimination between clutter and target-generated measurements is mathematically described as

$$\mathbf{Z}_k = \left(\mathbf{\Pi}_p \otimes \mathbf{I}_{N_z \times N_z}\right) \begin{bmatrix} \mathbf{Z}_k^c \\ \mathbf{Z}_k^t \end{bmatrix}, \tag{6}$$

where $\mathbf{Z}_k^c$ and $\mathbf{Z}_k^t$ are ordered vectors of clutter- and target-originated measurements, respectively, N_z is the dimension of the single-measurement space, and $\otimes$ denotes the Kronecker product. Further, $\mathbf{\Pi}_p$ is an M_k-dimensional random permutation matrix which models the uncertainty with respect to which measurements are generated by targets.

The uncertainty in the association between target-generated measurements and true targets, and the model for the corresponding target-generated measurements, are here described as

$$\mathbf{Z}_k^t = \big(\mathbf{C}_k \otimes \mathbf{I}_{N_z \times N_z}\big) h_k(\mathbf{x}_k) + \mathbf{w}_k, \tag{7}$$

where $\mathbf{C}_k$ is an unknown $M_t \times N$-dimensional matrix which determines the contribution of each target on each target-generated measurement. When sensor resolution problems are not present, a measurement can only originate from a single target. In that case, $\mathbf{C}_k$ has at most one single non-zero element per row (a one), which determines the originating target of that measurement. However, for situations with resolution limitations, several targets can be perceived as one by the sensor, and thus give rise to a joint/merged measurement. Then, the $\mathbf{C}_k$ matrix will have several non-zero elements in the corresponding row, where the size of each element describes the contribution from each target to that measurement. The multi-target observation function h_k transforms the joint state vector $\mathbf{x}_k$ to the measurement space, and the noise process $\mathbf{w}_k$ is assumed Gaussian with zero mean and block-diagonal covariance matrix $\mathbf{R}_k$. For linear measurement models, the target-generated measurements are given by

$$\mathbf{Z}_k^t = \big(\mathbf{C}_k \otimes \mathbf{I}_{N_z \times N_z}\big) \tilde{\mathbf{H}} \mathbf{x}_k + \mathbf{w}_k, \tag{8}$$

where

$$\tilde{\mathbf{H}} = \mathrm{diag}\big\{ \underbrace{\mathbf{H}, \cdots, \mathbf{H}}_{N \text{ times}} \big\}, \tag{9}$$

and where $\mathbf{H}$ is the single-target observation matrix. The measurement noise $\mathbf{w}_k$ is assumed independent of the process noise $\mathbf{v}_k$.

For the measurement model to be complete, we also need to model the properties of the clutter measurements. We here assume a spatially homogeneous Possion process. A single clutter measurement, $\mathbf{z}_k^c$, is hence distributed as

$$\mathbf{z}_k^c \sim \mathrm{Uniform}\big(\mathrm{FoV}\big), \tag{10}$$

where FoV is the field-of-view of the sensor, while the total number of clutter detections, N_c, is distributed according to

$$N_c \sim \text{Poisson}\big(\lambda \cdot |\text{FoV}|\big), \tag{11}$$

in which λ is the clutter intensity and $|\text{FoV}|$ the volume of the FoV. The algorithms described in this article, however, are not restricted to Poisson-distributed clutter.

3 Sensor resolution modeling framework for arbitrary target numbers

To be able to track targets under resolution limitations with high accuracy, we wish to create a model that

1. represents the probability that a group of targets is unresolved in a qualitatively correct way,
2. provides a reasonable description of the properties of a measurement from an unresolved group, and
3. is mathematically tractable and possible to incorporate in a Bayesian tracking framework.

In this section, we propose a novel framework for sensor resolution modeling for arbitrary, but known, number of targets, which fulfills the three requirements above. To complete the framework, models for the resolution probability of two targets and a measurement model for a group target are required. In Section 4, two such models for radar sensors are presented.

We start by defining a group target.

Definition 1 *When two or more targets lead to a single (joint/merged) detection, they are called a group target.*

The probability that a group target appears is large when the targets are closely spaced compared to the resolution capability of the sensor. Note that the definition is on a per-scan basis, so that groups can be formed and dissolved from scan to scan.

At a certain time instant, there are many different possibilities regarding which targets form an unresolved group and which are resolved. This is similar to the classic data association problem, where there are many different explanations regarding measurement-to-target associations. We thus make the following definition of a resolution event:

Definition 2 *A resolution event describes which targets form unresolved groups and which are resolved.*

So at a given time instant, many resolution events are possible, and we need a resolution model to provide information regarding the probabilities of those events.

Example 1 *Say that there are five targets present in the scene. An example of a resolution event,* $\mathcal{R}$*, is then:* $\mathcal{R} = \big\{\{1,2\},\{3\},\{4,5\}\big\}$*, meaning that targets* 1 *and* 2 *form an unresolved group, target* 3 *is resolved and targets* 4 *and* 5 *form an unresolved group.*

3.1 Resolution probabilities

To model a resolution event, we propose the use of a graph representation, where each node in the graph represents a target, and where an edge between nodes represent the event that those two targets are mutually unresolved. A group is then unresolved if there exists a walk through the corresponding target nodes. With this approach, a resolution event (cf. Definition 2) may correspond to several resolution graphs. For an example of a resolution graph, see Fig. 1. In this example, only one graph can be generated from the resolution event.

The edges of a resolution graph describes the pairwise interactions between the targets. For the calculation of the probability of a resolution graph, we make the following assumption:

Assumption 1 *The edge probabilities of a resolution graph are independent. That is, knowing that two targets are mutually unresolved provides no information regarding the probability that another pair of targets is unresolved.*

Figure 1: Illustration of a graph, $\mathcal{G}^1$, which describes the resolution event that target one is resolved, while targets two and three form an unresolved group.

For the calculation of the resolution graph probability we introduce the probability, P_u, according to the following definition.

Definition 3 *The probability that the pair of targets with states* $\mathbf{x}_k^{(i)}$ *and* $\mathbf{x}_k^{(j)}$ *are mutually unresolved is given by* $P_u\big(\mathbf{x}_k^{(i)}, \mathbf{x}_k^{(j)}\big)$. *Correspondingly, the probability that they are resolved is given by* $1 - P_u\big(\mathbf{x}_k^{(i)}, \mathbf{x}_k^{(j)}\big)$.

The probability of the graph $\mathcal{G}^1$, given by Fig. 1, is thus given by

$$\begin{aligned}\Pr\big\{\mathcal{G}^1\big|\mathbf{x}_k\big\} = P_u\big(\mathbf{x}_k^{(2)}, \mathbf{x}_k^{(3)}\big)\Big(1 - P_u\big(\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)}\big)\Big) \\ \times \Big(1 - P_u\big(\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(3)}\big)\Big). \qquad (12)\end{aligned}$$

Example 2 *Consider the resolution event* $\mathcal{R} = \big\{\{1,2,3\}\big\}$ *that the three targets present are all unresolved. There are exactly four graphs,* $\mathcal{G}^1$ *to* $\mathcal{G}^4$, *leading to this event, as illustrated in Fig. 2. The probability of the resolution event is then*

$$\begin{aligned}\Pr\big\{\mathcal{R}\big|\mathbf{x}_k\big\} = \Pr\big\{\mathcal{G}^1\big|\mathbf{x}_k\big\} + \Pr\big\{\mathcal{G}^2\big|\mathbf{x}_k\big\} \\ + \Pr\big\{\mathcal{G}^3\big|\mathbf{x}_k\big\} + \Pr\big\{\mathcal{G}^4\big|\mathbf{x}_k\big\},\end{aligned}$$

where the respective graph probabilities are calculated similarly to the graph probability in (12).

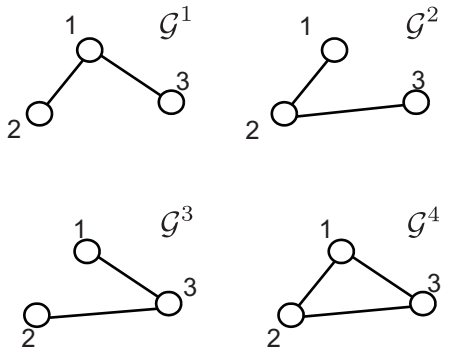

Figure 2: Illustration of the four graphs that lead to the resolution event $\mathcal{R}$, which describes the event that a group of three targets is unresolved.

From a resolution event, all possible resolution graphs that could be generated from the event are not always feasible, since some of them represent

cases which are not physically reasonable. An example of such a case is shown in Fig. 3. To circumvent generation of such events, a graph node can only be connected to its nearest neighbors in each measurement dimension, assuming that the resolution is independent between dimensions.

Definition 4 *A resolution graph where connections (edges) only exist between the nearest neighboring nodes in each measurement dimension is called feasible. Further, a graph which does not fulfill this is termed unfeasible.*

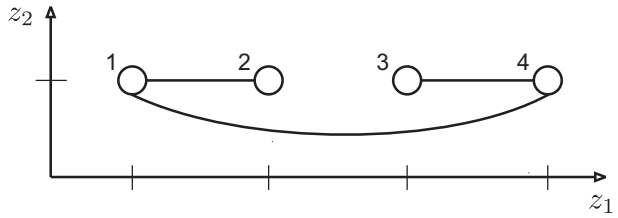

Figure 3: Example of an unfeasible graph. In the graph, the targets 1 and 4 are connected even though they are not nearest neighbors in either the z_1 or the z_2 dimension, which makes the graph unfeasible.

In Fig. 2 all graphs are feasible since connections only exist to the nearest neighbors in the measurement directions (being the horizontal and vertical axes). In Fig. 3, however, the graph is unfeasible since there exists a link between nodes 1 and 4, which are not the nearest neighbors in the z_1 dimension. That link thus results in the contradictory event that targets 1 and 4 are unresolved, while the more closely spaced targets 2 and 3 (in the same dimension) are resolved.

To find an expression for a general graph, $\mathcal{G}$, we let $\mathcal{S}_e$ be the set of pairwise targets that are connected with an edge in the graph, and $\mathcal{S}_0$ be the set of pairwise targets that are not connected. Then,

$$\Pr\left\{\mathcal{G}\middle|\mathbf{x}_k\right\} = \prod_{\mathcal{S}_e} P_u(\mathcal{S}_e) \prod_{\mathcal{S}_0} \big(1 - P_u(\mathcal{S}_0)\big). \tag{13}$$

Note that $\mathcal{S}_0$ and $\mathcal{S}_e$ are given by $\mathcal{G}$, and P_u is a function of the multi-target state $\mathbf{x}_k$. Further, we define the product over an empty set as one.

The procedure of obtaining all feasible graphs and their respective probabilities at a given time index is as follows:

1. Find all resolution events, $\mathcal{R}$.

2. For each resolution event, generate all feasible resolution graphs, $\mathcal{G}$.

3. Calculate the probability of each feasible graph using the resolution probabilities, P_u (see (12) for an example).

3.2 Graph likelihood

Apart from the probability of a graph, we would also like to describe the measurement model for a given graph $\mathcal{G}$, expressed as $p\big(\mathbf{Z}_k\big|\mathcal{G},\mathbf{d},\mathbf{x}_k\big)$. We call this model the *graph likelihood*. The data association vector $\mathbf{d}$ is included in the model since the distribution of the measurements depend on their origin. The vector is defined as

$$\mathbf{d} \triangleq \begin{bmatrix} d_1 & d_2 & \dots & d_{M_k} \end{bmatrix}^T, \tag{14}$$

where $d_j > 0$ if measurement j is assigned to the single target or group target d_j, and $d_j = 0$ if the measurement is due to clutter.

Conditioned on the data association, it is known which of the measurements in $\mathbf{Z}_k$ are target-generated and which originate from clutter. That is, the vectors $\mathbf{Z}_k^c$ (containing M_c observations) and $\mathbf{Z}_k^t$ (containing M_t observations) are known, and their distributions are given by $p\big(\mathbf{Z}_k^c\big|\mathcal{G},\mathbf{d},\mathbf{x}_k\big)$ and $p\big(\mathbf{Z}_k^t\big|\mathcal{G},\mathbf{d},\mathbf{x}_k\big)$. From the assumption of spatially uniform clutter, the first density is

$$p\big(\mathbf{Z}_k^c\big|\mathcal{G},\mathbf{d},\mathbf{x}_k\big) = \frac{1}{\big|\mathrm{FoV}\big|^{M_c}}. \tag{15}$$

Further, given the data association vector $\mathbf{d}$ and the graph $\mathcal{G}$, the target-generated measurements are distributed as

$$p\big(\mathbf{Z}_k^t\big|\mathcal{G},\mathbf{d},\mathbf{x}_k\big) = \prod_{i=1}^{M_t} p\big(\mathbf{z}_k^{t,(i)}\big|\mathbf{d},\mathcal{G},\mathbf{x}_k\big), \tag{16}$$

where $p\big(\mathbf{z}_k^{t,(i)}\big|\mathbf{d},\mathcal{G},\mathbf{x}_k\big)$ is the *group measurement model* (see Section 4.2 for one alternative). In (16), it is assumed that the measurements are independent, conditioned on their associations. Finally, the graph likelihood is given by

$$p\big(\mathbf{Z}_k\big|\mathcal{G},\mathbf{d},\mathbf{x}_k\big) = \frac{1}{\big|\mathrm{FoV}\big|^{M_c}} \prod_{i=1}^{M_t} p\big(\mathbf{z}_k^{t,(i)}\big|\mathbf{d},\mathcal{G},\mathbf{x}_k\big), \tag{17}$$

where M_c and M_t are known, given $\mathbf{d}$.

4 Models for pair-wise resolution probabilities and group measurements

In Section 3, a novel model structure was presented for handling sensor resolution modeling for an arbitrary, but known, number of targets. The model structure relies on a graph representation of a resolution event, where each resolution graph describes the pairwise interactions between the targets, under the resolution event.

The fundamental unit of the graph probabilities is the resolution probability $P_u\big(\mathbf{x}_k^{(i)}, \mathbf{x}_k^{(j)}\big)$, which states the probability that two targets with states $\mathbf{x}_k^{(i)}$and $\mathbf{x}_k^{(j)}$ are unresolved. To complete the resolution model we hence need a model for the probability P_u. Here, we consider radar sensors, and use the two-target resolution model by Koch and van Keuk [9], presented in Section 4.1. It should be noted that the proposed resolution framework of this article is not restricted to radar sensors, and that the described radar resolution model does not rely on the use of the two-target model in [9] as a basis for the resolution probability, but that it is used due to its appealing properties.

The resolution modeling framework also includes the graph likelihood $p\big(\mathbf{Z}_k\big|\mathcal{G}, \mathbf{d}, \mathbf{x}_k\big)$. The general expression for the likelihood is given in (17). To complete that likelihood, a group measurement model is needed, which provides an explicit expression of the target measurement density $p\big(\mathbf{z}_k^{t,(i)}\big|\mathcal{G}, \mathbf{d}, \mathbf{x}_k\big)$, conditioned on a graph, an association vector and a joint target state vector. In this article, we use a simple group measurement model which is presented in Section 4.2. However, the proposed resolution model does not hinge on that group measurement model, meaning that other group measurement models are possible to use instead.

4.1 Two-target resolution model

For a certain pair of targets $\mathbf{x}_k^{(i)}$, $\mathbf{x}_k^{(j)}$ in a set $\mathcal{S}$ in (13), the probability that they are unresolved is, according to [9],

$$P_u(\mathbf{x}_k^{(i)}, \mathbf{x}_k^{(j)}) = e^{-(\Delta\mathbf{r}^{i,j})^T(\mathbf{R}_{u,N_{\text{res}}})^{-1}\Delta\mathbf{r}^{i,j}}, \tag{18}$$

where N_{res} is the dimension of the measurement space (2 for range and azimuth), $\Delta\mathbf{r}^{i,j}$ is a vector with the distances between the predicted positions of targets i and j in the measurement space, and $\mathbf{R}_{u,N_{\text{res}}}$ is given by

$$\mathbf{R}_{u,N_{\text{res}}} = \frac{1}{\big(2\ln(2)\big)^{N_{\text{res}}/2}}\,\text{diag}\big\{\alpha_1^2, \ldots, \alpha_{N_{\text{res}}}^2\big\}. \tag{19}$$

The parameters α_1 to $\alpha_{N_{\text{res}}}$ in (19) describe the resolution capability of the sensor in the respective measurement dimensions. Note that the diagonal structure of $\mathbf{R}_{u,N_{\text{res}}}$ implies that the resolution in one dimension is independent of the resolutions in the other dimensions.

The probability $P_u(\mathbf{x}_k^{(i)}, \mathbf{x}_k^{(j)})$ can also be written as a scaled multivariate Gaussian

$$P_u(\mathbf{x}_k^{(i)}, \mathbf{x}_k^{(j)}) = \left|2\pi\mathbf{R}_{u,N_{\text{res}}}\right|^{1/2} \mathcal{N}\big(0; \Delta\mathbf{r}^{i,j}, \mathbf{R}_{u,N_{\text{res}}}\big). \tag{20}$$

Depending on the measurement model, the relation between $\Delta\mathbf{r}^{i,j}$ and the states $\mathbf{x}_k^{(i)}$ and $\mathbf{x}_k^{(j)}$ is either linear or non-linear.

4.2 Group measurement model

For an explicit expression of the graph likelihood $p\big(\mathbf{Z}_k \big| \mathcal{G}, \mathbf{d}, \mathbf{x}_k\big)$, we need a group measurement model which describes the properties of a merged measurement from an unresolved group. The assumed model states that a group measurement can be described as a measurement of the center of gravity in the measurement dimension. That is, for an unresolved group of n_g targets (possibly one), whose state vectors are gathered in the joint vector $\mathbf{x}_k^g$, their group measurement $\mathbf{z}_k^{t,(j)}$ is described by

$$\mathbf{z}_k^{t,(j)} = h_k^{n_g}\big(\mathbf{x}_k^g\big) + \mathbf{u}_k^{g,n_g}, \tag{21}$$

where $h_k^{n_g}\big(\mathbf{x}_k^g\big)$ provides the arithmetic mean of the group targets in the measurement dimension. For linear measurement models,

$$\mathbf{z}_k^{t,(j)} = \mathbf{H}_{n_g}\mathbf{x}_k^g + \mathbf{u}_k^{g,n_g}, \tag{22}$$

where

$$\mathbf{H}_{n_g} = \frac{1}{n_g}\underbrace{[\mathbf{H}, \cdots, \mathbf{H}]}_{n_g \text{ times}}. \tag{23}$$

The vector $\mathbf{u}_k^{g,n_g} \sim \mathcal{N}\big(\mathbf{0}, \mathbf{R}_k^{n_g}\big)$ models the measurement spread from an n_g-target group, which depends on the number of targets in the group and the measurement noise. Typically, the spread increases with the number of targets n_g, due to the radar target glint phenomenon.

Other group measurement models have been presented in the literature. A model for two targets, using amplitude information, is given in [3], and is simplified in [13], while a measurement model for automotive applications is proposed in [14].

5 Calculation of the posterior probability density function

In this section we describe the exact calculation of the posterior probability density function (pdf) $p\big(\mathbf{x}_k\big|\mathbf{Z}^k\big)$ under unknown resolution and data association events. We start by describing the general calculation of the density, and then describe how the different parts of the general expression are obtained.

To express the posterior density, we marginalize over the feasible resolution graphs, $\mathcal{G}$, and the set of data association hypotheses $\mathcal{D}(\mathcal{G})$ for each graph. By also using Bayes' rule and the Markov property, we obtain

$$p\big(\mathbf{x}_k\big|\mathbf{Z}^k\big) = \sum_{\mathcal{G}} \sum_{\mathbf{d}\in\mathcal{D}(\mathcal{G})} p\big(\mathbf{x}_k, \mathcal{G}, \mathbf{d}\big|\mathbf{Z}^k\big) \tag{24}$$

$$= \sum_{\mathcal{G}} \sum_{\mathbf{d}\in\mathcal{D}(\mathcal{G})} \frac{p\big(\mathbf{Z}_k\big|\mathcal{G}, \mathbf{d}, \mathbf{x}_k\big) p\big(\mathbf{x}_k, \mathcal{G}, \mathbf{d}\big|\mathbf{Z}^{k-1}\big)}{p\big(\mathbf{Z}_k\big|\mathbf{Z}^{k-1}\big)} \tag{25}$$

$$= \sum_{\mathcal{G}} \Pr\big\{\mathcal{G}\big|\mathbf{x}_k\big\} \times \sum_{\mathbf{d}\in\mathcal{D}(\mathcal{G})} \Pr\big\{\mathbf{d}\big|\mathcal{G}, \mathbf{x}_k\big\} \frac{p\big(\mathbf{Z}_k\big|\mathcal{G}, \mathbf{d}, \mathbf{x}_k\big) p\big(\mathbf{x}_k\big|\mathbf{Z}^{k-1}\big)}{p\big(\mathbf{Z}_k\big|\mathbf{Z}^{k-1}\big)} \tag{26}$$

where $p\big(\mathbf{x}_k\big|\mathbf{Z}^{k-1}\big)$ is the predicted density of the target states, $p\big(\mathbf{Z}_k\big|\mathcal{G}, \mathbf{d}, \mathbf{x}_k\big)$ is the graph likelihood (see (17)), $\Pr\big\{\mathbf{d}\big|\mathcal{G}, \mathbf{x}_k\big\}$ is the probability of a data association hypothesis $\mathbf{d}$, and $\Pr\big\{\mathcal{G}\big|\mathbf{x}_k\big\}$ is the graph probability, given by the resolution model (see (13) for a general expression).

As seen in (26), the calculation of the posterior density includes a *measurement update*, given by the second sum, and a *resolution model update*, given by the total expression. In the following sections, these calculations are described in more detail. Note that the discussion here does not make any assumptions on the models. For calculation under Gaussian assumptions, see Section 6.

5.1 Calculation of the predicted density

Given the prior density $p\big(\mathbf{x}_{k-1}\big|\mathbf{Z}^{k-1}\big)$, we calculate the predicted density $p\big(\mathbf{x}_k\big|\mathbf{Z}^{k-1}\big)$ by means of marginalization over the previous state $\mathbf{x}_{k-1}$,

$$p\big(\mathbf{x}_k\big|\mathbf{Z}^{k-1}\big) = \int p\big(\mathbf{x}_k, \mathbf{x}_{k-1}\big|\mathbf{Z}^{k-1}\big)d\mathbf{x}_{k-1} \tag{27}$$

$$= \int p\big(\mathbf{x}_k\big|\mathbf{x}_{k-1}, \mathbf{Z}^{k-1}\big)p\big(\mathbf{x}_{k-1}\big|\mathbf{Z}^{k-1}\big)d\mathbf{x}_{k-1} \tag{28}$$

$$= \int p\big(\mathbf{x}_k\big|\mathbf{x}_{k-1}\big)p\big(\mathbf{x}_{k-1}\big|\mathbf{Z}^{k-1}\big)d\mathbf{x}_{k-1}, \tag{29}$$

where we in the final step assume that the prediction process has the Markov property. In the prediction step, the prior density is propagated through the process model $p\big(\mathbf{x}_k\big|\mathbf{x}_{k-1}\big)$.

5.2 Measurement update

The measurement update includes the calculation of the data association probabilities under a resolution graph and the received data set, and the update of the predicted density function with the measurement likelihood. We start with the data association hypotheses.

5.2.1 Data association probabilities

We here state an expression for the conditional data association hypothesis probability $\Pr\big\{\mathbf{d}\big|\mathcal{G}, \mathbf{x}_k\big\}$ in (26). Using the definition of the data association vector $\mathbf{d}$ in (14),

$$\Pr\big\{\mathbf{d}\big|\mathcal{G}, \mathbf{x}_k\big\} = P_c(M_c)\frac{\big(M_k - M_t\big)!}{M_k!} \times \prod_{\{j:\,\mathbf{d}(j)=0\}}\big(1 - P_D^j\big) \prod_{\{j:\,\mathbf{d}(j)>0\}} P_D^j, \tag{30}$$

where $\{j : \mathbf{d}(j) = 0\}$ is the set of clutter detections, $\{j : \mathbf{d}(j) > 0\}$ is the set of target-generated measurements, and $P_c(M_c)$ is the probability of receiving M_c clutter measurements, which is given by the Poisson mass function with parameter $\lambda \cdot |\text{FoV}|$. Further, the detection probability P_D^j for measurement index j is the group detection probability, which can be modeled to attain different values for different number of targets in the group. Thus, we here assume that the detection probability only depends on the number of targets in the group, and not on their states.

5.2.2 Density update

The second part of the measurement update is to refine the predicted density $p\big(\mathbf{x}_k\big|\mathbf{Z}^{k-1}\big)$ with information from the current measurements, $\mathbf{Z}_k$, under the graph, $\mathcal{G}$ and data association $\mathbf{d} \in \mathcal{D}(\mathcal{G})$. That is, we seek an expression for the product $p\big(\mathbf{Z}_k\big|\mathcal{G},\mathbf{d},\mathbf{x}_k\big)p\big(\mathbf{x}_k\big|\mathbf{Z}^{k-1}\big)$, ignoring the scaling $p\big(\mathbf{Z}_k\big|\mathbf{Z}^{k-1}\big)$.

A general measurement model, under the assumption of uniformly distributed clutter in the measurement space, is given by (17). Using that model, the density product is

$$\begin{aligned} &p\big(\mathbf{Z}_k\big|\mathcal{G},\mathbf{d},\mathbf{x}_k\big)p\big(\mathbf{x}_k\big|\mathbf{Z}^{k-1}\big) \\ &\quad= \frac{1}{\big|\mathrm{FoV}\big|^{M_c}}\prod_{i=1}^{M_t} p\big(\mathbf{z}_k^{t,(i)}\big|\mathcal{G},\mathbf{d},\mathbf{x}_k\big)p\big(\mathbf{x}_k\big|\mathbf{Z}^{k-1}\big). \end{aligned} \tag{31}$$

The calculation of the measurement-updated pdf depends on the measurement model $p\big(\mathbf{Z}_k\big|\mathcal{G},\mathbf{d},\mathbf{x}_k\big)$. If it is linear and Gaussian, the first two moments of the updated density is calculated by the Kalman filter [15], for each target group. For non-linear models, the Extended Kalman filter (EKF) [16] or the Unscented Kalman filter (UKF) [17] can be used. More details about explicit expressions under Gaussian assumptions are found in Section 6. The calculation also depends on the predicted density $p\big(\mathbf{x}_k\big|\mathbf{Z}^{k-1}\big)$.

5.3 Update with the resolution model

As seen in (26), the update with the measurement model gives an increase in the number of density components, for each graph. In the update with the resolution model, each such component is multiplied with the graph probability. For a certain graph $\mathcal{G}$ and data association hypothesis $\mathbf{d}$ we thus make the update

$$\begin{aligned} &\Pr\big\{\mathcal{G}\big|\mathbf{x}_k\big\}\frac{1}{\big|\mathrm{FoV}\big|^{M_c}}\prod_{i=1}^{M_t} p\big(\mathbf{z}_k^{t,(i)}\big|\mathcal{G},\mathbf{d},\mathbf{x}_k\big)p\big(\mathbf{x}_k\big|\mathbf{Z}^{k-1}\big) \\ &\quad= \prod_{\mathcal{S}_e} P_u(\mathcal{S}_e)\prod_{\mathcal{S}_0}\big(1-P_u(\mathcal{S}_0)\big) \\ &\qquad\times\frac{1}{\big|\mathrm{FoV}\big|^{M_c}}\prod_{i=1}^{M_t} p\big(\mathbf{z}_k^{t,(i)}\big|\mathcal{G},\mathbf{d},\mathbf{x}_k\big)p\big(\mathbf{x}_k\big|\mathbf{Z}^{k-1}\big). \end{aligned} \tag{32}$$

Due to the $1-P_u$ factors, the resolution update further increases the number of density mixture components. To get a clearer insight into the

update calculations needed, we study the update in more detail, focusing on

$$\prod_{\mathcal{S}_e} P_u(\mathcal{S}_e) \prod_{\mathcal{S}_0} \big(1 - P_u(\mathcal{S}_0)\big) = \prod_{\mathcal{S}_e} P_u(\mathcal{S}_e) \big(1 - P_u(\mathcal{S}_0(1)\big) \\ \times \big(1 - P_u(\mathcal{S}_0(2)\big) \times \cdots \times \big(1 - P_u(\mathcal{S}_0(|\mathcal{S}_0|)\big). \tag{33}$$

In (33), $|\mathcal{S}_0|$ is the cardinality of the set $\mathcal{S}_0$, i.e., the number of resolved pairs in the graph $\mathcal{G}$. The $(1 - P_u)$ factors can be split up, leading to

$$\prod_{\mathcal{S}_e} P_u(\mathcal{S}_e) \prod_{\mathcal{S}_0} \big(1 - P_u(\mathcal{S}_0)\big) = \prod_{\mathcal{S}_e} P_u(\mathcal{S}_e) \\ \times \left(1 - \sum_{j=1}^{|\mathcal{S}_0|} P_u(\mathcal{S}_0(j)) + \sum_{j=1}^{|\mathcal{S}_0|} \sum_{\substack{l=1 \\ l \neq j}}^{|\mathcal{S}_0|} P_u(\mathcal{S}_0(j)) P_u(\mathcal{S}_0(l)) \right. \\ \left. - \ldots + \sum_{j=1}^{|\mathcal{S}_0|} \sum_{\substack{l=1 \\ l \neq j}}^{|\mathcal{S}_0|} \cdots \prod_{w=1}^{|\mathcal{S}_0|} P_u(\mathcal{S}_0(w)) \right). \tag{34}$$

As seen in (18), the resolution probabilities $P_u(\mathcal{S}_0(j))$ depend on the state vector $\mathbf{x}_k$. Hence, the multiplication of the measurement-updated density with a resolution probability does not only scale the density, but also affects its shape. The multiplication can thus be seen as a density update. Hence, for each term in (34), we can perform an update of the measurement-updated density. Each update can either be performed sequentially or in a single step, as described further in Section 6. The set of components generated by a resolution update, for a graph $\mathcal{G}$ and data associaiton hypothesis $\mathbf{d}$, are gathered in the set $\mathcal{U}(\mathcal{G})$.

6 Gaussian mixture approximation

In this section, we seek to find a Gaussian-mixture expression of the posterior density $p\big(\mathbf{x}_k \big| \mathbf{Z}^k\big)$. For the sake of convenience, we assume that the prior density $p\big(\mathbf{x}_{k-1} \big| \mathbf{Z}^{k-1}\big)$ is a single Gaussian, i.e.,

$$p\big(\mathbf{x}_{k-1} \big| \mathbf{Z}^{k-1}\big) = \mathcal{N}\big(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}\big). \tag{35}$$

The calculation will concern linear process and measurement models (cf. (3), (4), (21), and (22)) with Gaussian noise. The generalization to general models can be performed by linearization, similar to the EKF, or by approximations with the Unscented Transform [17]. For the probability that two targets are unresolved, P_u, we use the model in (20).

6.1 Prediction step

The prediction is given by (cf. (29))

$$p\big(\mathbf{x}_k\big|\mathbf{Z}^{k-1}\big) = \int p\big(\mathbf{x}_k\big|\mathbf{x}_{k-1}\big)p\big(\mathbf{x}_{k-1}\big|\mathbf{Z}^{k-1}\big)d\mathbf{x}_{k-1} \tag{36}$$

$$= \mathcal{N}\big(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}\big), \tag{37}$$

where

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1|k-1} \tag{38}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_k \tag{39}$$

describe the predicted mean value and covariance matrix, respectively.

6.2 Measurement update

In the measurement update step, the predicted density, $p\big(\mathbf{x}_k\big|\mathbf{Z}^{k-1}\big)$, is updated with information from the current measurement set, $\mathbf{Z}_k$. The update relies on the single-target and group-target measurement models. A general expression for the measurement update is given by (31). By assuming that target groups are independent, the predicted density can be split into a product,

$$p\big(\mathbf{x}_k\big|\mathbf{Z}^{k-1}\big) = \prod_{j=1}^{N_g} p\big(\mathbf{x}_k^{g_j}\big|\mathbf{Z}^{k-1}\big), \tag{40}$$

where g_j is the group index and N_g is the number of groups. Note that the group size, here, can be equal to one.

The measurement update step thus involves the calculation of

$$p\big(\mathbf{Z}_k\big|\mathcal{G}, \mathbf{d}, \mathbf{x}_k\big)p\big(\mathbf{x}_k\big|\mathbf{Z}^{k-1}\big)$$

$$= \frac{1}{\big|\mathrm{FoV}\big|^{M_c}} \prod_{i=1}^{M_t} p\big(\mathbf{z}_k^{t,(i)}\big|\mathcal{G}, \mathbf{d}, \mathbf{x}_k\big) \prod_{j=1}^{N_g} p\big(\mathbf{x}_k^{g_j}\big|\mathbf{Z}^{k-1}\big). \tag{41}$$

Due to the independence assumption, the update can be performed group by group. We will describe the expressions for one such update,

$$p\big(\mathbf{z}_k^{t,(i)}\big|\mathcal{G}, \mathbf{d}, \mathbf{x}_k\big)p\big(\mathbf{x}_k^{g_j}\big|\mathbf{Z}^{k-1}\big),$$

of group g_j with its associated measurement $\mathbf{z}_k^{t,(i)}$. Using the group measurement model in (22), the update is given by

$$\begin{aligned} &p\big(\mathbf{z}_k^{t,(i)}\big|\mathcal{G},\mathbf{d},\mathbf{x}_k\big)p\big(\mathbf{x}_k^{g_j}\big|\mathbf{Z}^{k-1}\big) \\ &\quad=\mathcal{N}\big(\mathbf{z}_k^{t,(i)};\mathbf{H}_{n_{g_j}}\mathbf{x}_k^{g_j},\mathbf{R}_k^{n_{g_j}}\big)\mathcal{N}\big(\mathbf{x}_k^{g_j};\hat{\mathbf{x}}_{k|k-1}^{g_j},\mathbf{P}_{k|k-1}^{g_j}\big). \end{aligned} \tag{42}$$

By using the following property of a Gaussian product

$$\mathcal{N}\big(\mathbf{x};\hat{\mathbf{x}},\mathbf{P}\big)\mathcal{N}\big(\mathbf{z};\mathbf{H}\mathbf{x},\mathbf{R}\big)=\mathcal{N}\big(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}\big)\mathcal{N}\big(\mathbf{z};\hat{\mathbf{z}},\mathbf{S}\big), \tag{43}$$

where

$$\hat{\mathbf{z}}=\mathbf{H}\hat{\mathbf{x}} \tag{44}$$

$$\mathbf{S}=\mathbf{H}\mathbf{P}\mathbf{H}^T+\mathbf{R} \tag{45}$$

$$\boldsymbol{\mu}=\hat{\mathbf{x}}+\mathbf{K}\big(\mathbf{z}-\mathbf{H}\hat{\mathbf{x}}\big) \tag{46}$$

$$\boldsymbol{\Sigma}=\big(\mathbf{I}-\mathbf{K}\mathbf{H}\big)\mathbf{P} \tag{47}$$

$$\mathbf{K}=\mathbf{P}\mathbf{H}^T\mathbf{S}^{-1}, \tag{48}$$

we obtain

$$\begin{aligned} &\mathcal{N}\big(\mathbf{z}_k^{t,(i)};\mathbf{H}_{n_{g_j}}\mathbf{x}_k^{g_j},\mathbf{R}_k^{n_{g_j}}\big)\mathcal{N}\big(\mathbf{x}_k^{g_j};\hat{\mathbf{x}}_{k|k-1}^{g_j},\mathbf{P}_{k|k-1}^{g_j}\big) \\ &\quad=\mathcal{N}\big(\mathbf{z}_k^{t,(i)};\hat{\mathbf{z}}_k^{g_j},\mathbf{S}^{g_j}\big)\mathcal{N}\big(\mathbf{x}_k^{g_j};\hat{\mathbf{x}}_{k|k}^{g_j},\mathbf{P}_{k|k}^{g_j}\big), \end{aligned} \tag{49}$$

where the mean values and covariance matrices are given by identification from (43)–(48).

The update can also be performed in a single step. For that we utilize the multi-target measurement model in (8). Thereby,

$$\begin{aligned} &p\big(\mathbf{Z}_k\big|\mathcal{G},\mathbf{d},\mathbf{x}_k\big)p\big(\mathbf{x}_k\big|\mathbf{Z}^{k-1}\big) \\ &\quad=\mathcal{N}\big(\mathbf{Z}_k;\breve{\mathbf{H}}\mathbf{x}_k,\mathbf{R}\otimes\mathbf{I}_{M_t\times M_t}\big)\mathcal{N}\big(\mathbf{x}_k;\hat{\mathbf{x}}_{k|k-1},\mathbf{P}_{k|k-1}\big) &(50)\\ &\quad=\mathcal{N}\big(\mathbf{Z}_k;\breve{\mathbf{H}}\hat{\mathbf{x}}_{k|k-1},\mathbf{S}^{\mathcal{G},\mathbf{d}}\big)\mathcal{N}\big(\mathbf{x}_k;\hat{\mathbf{x}}_{k|k},\mathbf{P}_{k|k}\big), &(51)\end{aligned}$$

where

$$\breve{\mathbf{H}}=\mathbf{C}_k^{\mathcal{G},\mathbf{d}}\otimes\mathbf{I}_{N_z\times N_z}\tilde{\mathbf{H}} \tag{52}$$

describes the relation between target states and the joint measurement vector, in which $\mathbf{C}_k^{\mathcal{G},\mathbf{d}}$ represents the current resolution and data association events. By identification from (43)–(48),

$$\hat{\mathbf{Z}}_k^{\mathcal{G},\mathbf{d}}=\breve{\mathbf{H}}\hat{\mathbf{x}}_{k|k-1} \tag{53}$$

$$\mathbf{S}^{\mathcal{G},\mathbf{d}}=\breve{\mathbf{H}}\mathbf{P}_{k|k-1}\breve{\mathbf{H}}^T \tag{54}$$

$$\mathbf{K}^{\mathcal{G},\mathbf{d}}=\mathbf{P}_{k|k-1}\breve{\mathbf{H}}^T(\mathbf{S}^{\mathcal{G},\mathbf{d}})^{-1} \tag{55}$$

$$\hat{\mathbf{x}}_{k|k}=\hat{\mathbf{x}}_{k|k-1}+\mathbf{K}^{\mathcal{G},\mathbf{d}}\big(\mathbf{Z}_k-\breve{\mathbf{H}}\hat{\mathbf{x}}_{k|k-1}\big). \tag{56}$$

6.3 Sequential update with the resolution model

As seen in (32) and (34), the resolution model update involves the calculation of the product between a measurement-updated density and a sum of products. The update can be performed summand by summand, producing a density mixture. For each summand, the updated density is multiplied with a product of P_u factors. This calculation can either be carried out sequentially, factor by factor, or in a single step. Here we describe the sequential update.

Let us start with a single P_u factor, $P_u\big(\mathbf{x}_k^{(i)}, \mathbf{x}_k^{(j)}\big)$. From (51), the updated density is described by a product of two Gaussian densities, where only one depends on $\mathbf{x}_k$. The resolution model update is thus

$$\begin{aligned} &P_u\big(\mathbf{x}_k^{(i)}, \mathbf{x}_k^{(j)}\big)\mathcal{N}\big(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}\big) \\ &\quad = \big|2\pi\mathbf{R}_{u,N_{\text{res}}}\big|^{1/2}\mathcal{N}\big(0; \Delta\mathbf{r}^{i,j}, \mathbf{R}_{u,N_{\text{res}}}\big)\mathcal{N}\big(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}\big), \end{aligned} \tag{57}$$

where

$$\Delta\mathbf{r}^{i,j} = \mathbf{H}\big(\mathbf{x}_k^{(i)} - \mathbf{x}_k^{(j)}\big). \tag{58}$$

To describe the product of Gaussians, we would like to express the resolution model Gaussian as a function of the joint target state vector $\mathbf{X}_k$. To do so, we use the Kronecker delta

$$\delta_{k,i} \triangleq \begin{cases} 1 & \text{if } k = i \\ 0 & \text{otherwise,} \end{cases} \tag{59}$$

the $1 \times N$ vector

$$\pi^{(i,j)} \triangleq \big[\delta_{1,i} - \delta_{1,j}, \ldots, \delta_{N,i} - \delta_{N,j}\big] \tag{60}$$

and the matrix

$$\mathbf{\Pi}^{(i,j)} = \pi^{(i,j)} \otimes \mathbf{I}_{N_z}. \tag{61}$$

Then,

$$\Delta\mathbf{r}^{i,j} = \mathbf{\Pi}^{(i,j)}\tilde{\mathbf{H}}\mathbf{x}_k, \tag{62}$$

where $\tilde{\mathbf{H}}$ is defined in (9).

The update with one P_u factor is hence

$$P_u\big(\mathbf{x}_k^{(i)}, \mathbf{x}_k^{(j)}\big)\mathcal{N}\big(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}\big) = \big|2\pi\mathbf{R}_{u,N_{\text{res}}}\big|^{1/2} \tag{63}$$

$$\times\, \mathcal{N}\big(\mathbf{0}; \mathbf{\Pi}^{(i,j)}\tilde{\mathbf{H}}\mathbf{x}_k, \mathbf{R}_{u,N_{\text{res}}}\big)\mathcal{N}\big(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}\big) \tag{64}$$

when the measurement model is linear. This can be re-written according to the Gaussian product formula as

$$\begin{aligned}&\mathcal{N}\big(\mathbf{0};\boldsymbol{\Pi}^{(i,j)}\tilde{\mathbf{H}}\mathbf{x}_k,\mathbf{R}_{u,N_{\text{res}}}\big)\mathcal{N}\big(\mathbf{x}_k;\hat{\mathbf{x}}_{k|k},\mathbf{P}_{k|k}\big)\\&\quad=\mathcal{N}\big(\mathbf{0};\boldsymbol{\Pi}^{(i,j)}\tilde{\mathbf{H}}\hat{\mathbf{x}}_{k|k},\mathbf{S}^{i,j}\big)\mathcal{N}\big(\mathbf{x}_k;\hat{\mathbf{x}}^{i,j}_{k|k},\mathbf{P}^{i,j}_{k|k}\big),\end{aligned}\tag{65}$$

where $\hat{\mathbf{x}}^{i,j}_{k|k}$ and $\mathbf{P}^{i,j}_{k|k}$ describe the expected value and the covariance matrix of the target states given measurement update under hypothesis $\mathbf{d}$, for graph $\mathcal{G}$, and resolution model update for the unresolved target pair (i,j). The exact form of $\mathbf{S}^{i,j}$, $\hat{\mathbf{x}}^{i,j}_{k|k}$ and $\mathbf{P}^{i,j}_{k|k}$ are given by identification from (43)–(48). As seen in the update equation, the 'negative information' from a missed detection due to resolution limitations is hence incorporated in the tracking framework as a measured '0' of the separation of the target pair.

When the summands in (34) involves several P_u factors, similar calculations as in (64) are performed sequentially, factor by factor. That is, if we have a second factor $P_u\big(\mathbf{x}^{(l)}_k,\mathbf{x}^{(m)}_k\big)$, we perform a similar calculation of

$$P_u\big(\mathbf{x}^{(l)}_k,\mathbf{x}^{(m)}_k\big)\mathcal{N}\big(\mathbf{x}_k;\hat{\mathbf{x}}^{i,j}_{k|k},\mathbf{P}^{i,j}_{k|k}\big),$$

and so on. In general (cf. (34)),

$$\begin{aligned}&\prod_{w=1}^{|\mathcal{S}_0|}P_u(\mathcal{S}_0(w)\mathcal{N}\big(\mathbf{x}_k;\hat{\mathbf{x}}_{k|k},\mathbf{P}_{k|k}\big)=\big|2\pi\mathbf{R}_{u,N_{\text{res}}}\big|^{|\mathcal{S}_0|/2}\\&\times\prod_{w=1}^{|\mathcal{S}_0|}\mathcal{N}\big(\mathbf{0};\boldsymbol{\Pi}^{(w)}\tilde{\mathbf{H}}\hat{\mathbf{x}}^{w-1}_{k|k},\mathbf{S}^{w}_{k|k}\big)\mathcal{N}\big(\mathbf{x}_k;\hat{\mathbf{x}}^{\mathcal{S}_0}_{k|k},\mathbf{P}^{\mathcal{S}_0}_{k|k}\big),\end{aligned}\tag{66}$$

where $\hat{\mathbf{x}}^{w-1}$ is the state estimate after the update with resolution pair $w-1$. Further, $\mathbf{S}^w$ depends on the previous covariance $\mathbf{P}^{w-1}_{k|k}$ (cf. (45)), where we define $\hat{\mathbf{x}}^0_{k|k}\triangleq\hat{\mathbf{x}}_{k|k}$, and where $\hat{\mathbf{x}}^{\mathcal{S}_0}_{k|k}$ and $\mathbf{P}^{\mathcal{S}_0}_{k|k}$ are the state estimate and covariance matrix after the sequence of updates, for the set $\mathcal{S}_0$.

6.4 One-step update with the resolution model

Instead of making a sequential resolution update, as in the previous section, we can make a single update for each summand in (34). To do so, we first

note that the product of P_u factors (cf. (34)) can be written as

$$\prod_{w=1}^{|\mathcal{S}_0|} P_u(\mathcal{S}_0(w)) = e^{-(\Delta\mathbf{r}^{i,j})^T(\mathbf{R}_{u,N_{\text{res}}})^{-1}\Delta\mathbf{r}^{i,j}} \tag{67}$$

$$= \prod_{w=1}^{|\mathcal{S}_0|} e^{-(\mathbf{\Pi}^{(i,j)}\tilde{\mathbf{H}}\mathbf{x}_k)^T(\mathbf{R}_{u,N_{\text{res}}})^{-1}\mathbf{\Pi}^{(i,j)}\tilde{\mathbf{H}}\mathbf{x}_k} \tag{68}$$

$$= e^{-\frac{1}{2}\sum_{w=1}^{|\mathcal{S}_0|}(\mathbf{\Pi}^{(i,j)}\tilde{\mathbf{H}}\mathbf{x}_k)^T(\mathbf{R}_{u,N_{\text{res}}})^{-1}\mathbf{\Pi}^{(i,j)}\tilde{\mathbf{H}}\mathbf{x}_k} \tag{69}$$

$$= e^{-\frac{1}{2}\mathbf{x}_k^T\left\{\sum_{w=1}^{|\mathcal{S}_0|}(\mathbf{\Pi}^{(i,j)}\tilde{\mathbf{H}})^T(\mathbf{R}_{u,N_{\text{res}}})^{-1}\mathbf{\Pi}^{(i,j)}\tilde{\mathbf{H}}\right\}\mathbf{x}_k} \tag{70}$$

$$= e^{-\frac{1}{2}\mathbf{x}_k^T\tilde{\mathbf{R}}_u^{-1}\mathbf{x}_k}. \tag{71}$$

By using the Kronecker product,

$$\tilde{\mathbf{R}}_u^{-1} \triangleq \sum_{w=1}^{|\mathcal{S}_0|}(\mathbf{\Pi}^{(i,j)}\tilde{\mathbf{H}})^T(\mathbf{R}_{u,N_{\text{res}}})^{-1}\mathbf{\Pi}^{(i,j)}\tilde{\mathbf{H}} \tag{72}$$

$$= \sum_{w=1}^{|\mathcal{S}_0|}\tilde{\mathbf{H}}^T\Big\{\big(\pi^{(i,j)}\big)^T\pi^{(i,j)}\big)\otimes\mathbf{R}_{u,N_{\text{res}}}^{-1}\Big\}\tilde{\mathbf{H}}. \tag{73}$$

The matrices $\tilde{\mathbf{H}}$ and $\mathbf{R}_{u,N_{\text{res}}}$ do not depend on i and j and can hence be moved out of the summation. So, by defining

$$\mathbf{G} \triangleq \sum_{w=1}^{|\mathcal{S}_0|}\big(\pi^{(i,j)}\big)^T\pi^{(i,j)}\big), \tag{74}$$

we get

$$\tilde{\mathbf{R}}_u^{-1} = \tilde{\mathbf{H}}^T\big(\mathbf{G}\otimes\mathbf{R}_{u,N_{\text{res}}}^{-1}\big)\tilde{\mathbf{H}}. \tag{75}$$

Since $\mathbf{G}$ is positive semi-definite, it has a matrix square root, $\mathbf{D}$, and we can thus write

$$\mathbf{G} = \mathbf{D}^T\mathbf{D}. \tag{76}$$

Then, repeatedly using the Kronecker product rule [18]

$$\big(\mathbf{A}\otimes\mathbf{B}\big)\big(\mathbf{C}\otimes\mathbf{D}\big) = \big(\mathbf{AC}\big)\otimes\big(\mathbf{BD}\big), \tag{77}$$

we have

$$\mathbf{G}\otimes\mathbf{R}_{u,N_{\text{res}}}^{-1} = \big(\mathbf{D}^T\mathbf{D}\big)\otimes\mathbf{R}_{u,N_{\text{res}}}^{-1} \tag{78}$$

$$= \big(\mathbf{D}^T\otimes\mathbf{R}_{u,N_{\text{res}}}^{-1}\big)\big(\mathbf{D}\otimes\mathbf{I}_{N_{\text{resr}}}\big) \tag{79}$$

$$= \big(\mathbf{D}^T\otimes\mathbf{I}_{N_{\text{res}}}\big)\big(\mathbf{I}_N\otimes\mathbf{R}_{u,N_{\text{res}}}^{-1}\big)\big(\mathbf{D}\otimes\mathbf{I}_{N_{\text{res}}}\big). \tag{80}$$

Since

$$\mathbf{D}^T \otimes \mathbf{I}_{N_{\text{res}}} = \mathbf{D}^T \otimes \mathbf{I}_{N_{\text{res}}}^T = \big(\mathbf{D} \otimes \mathbf{I}_{N_{\text{res}}}\big)^T, \tag{81}$$

the matrix $\tilde{\mathbf{R}}_u^{-1}$ can be written as

$$\tilde{\mathbf{R}}_u^{-1} = \tilde{\mathbf{H}}^T \big(\mathbf{D} \otimes \mathbf{I}_{N_{\text{res}}}\big)^T \big(\mathbf{I}_N \otimes \mathbf{R}_{u,N_{\text{res}}}^{-1}\big)\big(\mathbf{D} \otimes \mathbf{I}_{N_{\text{res}}}\big)\tilde{\mathbf{H}}. \tag{82}$$

So, through

$$\begin{aligned}\mathbf{x}_k^T \tilde{\mathbf{R}}_u^{-1} \mathbf{x}_k &= \mathbf{x}_k^T \tilde{\mathbf{H}}^T \big(\mathbf{D} \otimes \mathbf{I}_{N_{\text{res}}}\big)^T \\ &\qquad \times \big(\mathbf{I}_N \otimes \mathbf{R}_{u,N_{\text{res}}}^{-1}\big)\big(\mathbf{D} \otimes \mathbf{I}_{N_{\text{res}}}\big)\tilde{\mathbf{H}}\mathbf{x}_k \qquad (83)\\ &= \big((\mathbf{D} \otimes \mathbf{I}_{N_{\text{res}}})\tilde{\mathbf{H}}\mathbf{x}_k\big)^T \big(\mathbf{I}_N \otimes \mathbf{R}_{u,N_{\text{res}}}^{-1}\big)\big((\mathbf{D} \otimes \mathbf{I}_{N_{\text{res}}})\tilde{\mathbf{H}}\mathbf{x}\big), \qquad (84)\end{aligned}$$

the probability in (71) is described. By further using that

$$\big(\mathbf{I}_N \otimes \mathbf{R}_{u,N_{\text{res}}}^{-1}\big) = \big(\mathbf{I}_N \otimes \mathbf{R}_{u,N_{\text{res}}}\big)^{-1} \tag{85}$$

the product of P_u factors can be written as a scaled Gaussian density

$$\prod_{w=1}^{|\mathcal{S}_0|} P_u(\mathcal{S}_0(w) = \big|2\pi\mathbf{I}_N \otimes \mathbf{R}_{u,N_{\text{res}}}\big| \mathcal{N}\big(\mathbf{0}; (\mathbf{D} \otimes \mathbf{I}_{N_{\text{res}}})\tilde{\mathbf{H}}\mathbf{x}_k, \mathbf{I}_N \otimes \mathbf{R}_{u,N_{\text{res}}}\big). \tag{86}$$

Using the description in (86), the update of the measurement-updated density with one of the summation terms in (34) can be performed in a single Gaussian-product step (the updated density, is for example given by (51))

$$\begin{aligned}&\mathcal{N}\big(\mathbf{0}; (\mathbf{D} \otimes \mathbf{I}_{N_{\text{res}}})\tilde{\mathbf{H}}\mathbf{x}_k, \mathbf{I}_N \otimes \mathbf{R}_{u,N_{\text{res}}}\big)\mathcal{N}\big(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}\big) \\ &\quad = \mathcal{N}\big(\mathbf{0}; (\mathbf{D} \otimes \mathbf{I}_{N_{\text{res}}})\tilde{\mathbf{H}}\hat{\mathbf{x}}_{k|k}, \mathbf{S}^u\big)\mathcal{N}\big(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^u, \mathbf{P}_{k|k}^u\big), \qquad (87)\end{aligned}$$

where

$$\begin{aligned}\mathbf{S}^u &= \big((\mathbf{D} \otimes \mathbf{I}_{N_{\text{res}}})\tilde{\mathbf{H}}\big)^T \mathbf{P}_{k|k}\big((\mathbf{D} \otimes \mathbf{I}_{N_{\text{res}}})\tilde{\mathbf{H}}\big) \\ &\qquad + \mathbf{I}_N \otimes \mathbf{R}_{u,N_{\text{res}}} \qquad (88)\\ \mathbf{K}^u &= \mathbf{P}_{k|k}\big((\mathbf{D} \otimes \mathbf{I}_{N_{\text{res}}})\tilde{\mathbf{H}}\big)^T (\mathbf{S}^u)^{-1} \qquad (89)\\ \hat{\mathbf{x}}_{k|k} &= \mathbf{x}_{k|k} + \mathbf{K}^u\Big(\mathbf{0} - \big(\mathbf{D} \otimes \mathbf{I}_{N_{\text{res}}}\big)\tilde{\mathbf{H}}\hat{\mathbf{x}}_{k|k}\Big) \qquad (90)\\ \mathbf{P}_{k|k}^u &= \Big(\mathbf{I} - \mathbf{K}^u\big(\mathbf{D} \otimes \mathbf{I}_{N_{\text{res}}}\big)\tilde{\mathbf{H}}\Big)\mathbf{P}_{k|k}. \qquad (91)\end{aligned}$$

The one-step update described here is performed for each term in the sums over $\big|\mathcal{S}_0\big|$ in (34). The resolution model update is then finalized by the update with the product of $P_u(\mathcal{S}_e)$ factors, which can also be performed in a single calculation for each term.

6.5 Summary

The update with measurement and resolution models can be performed in four different ways:

1. Sequential measurement and resolution updates.
2. Sequential measurement update and one-step resolution update.
3. One-step measurement update and sequential resolution update.
4. One-step measurement and resolution updates.

The advantage with the one-step updates is that they keep the form of the Gaussian mixture throughout the updates, since no assumption about independence between target groups is necessary. The drawback, however, is that the calculations require high-dimensional matrix operations. Then, the sequential update is a further approximation which makes it faster but leads to the effect that the order matters and that the graph feasibility might not be preserved.

7 Joint Probabilistic Data Association filtering using the resolution model

In Section 6, the calculation of the posterior density was described under Gaussian assumptions. If the prior density is Gaussian, the posterior density is a Gaussian mixture, where the number of components depend on

- the number of resolution graphs, $\mathcal{G}$,
- the number of open links in the graphs, yielding $\left(1-P_u\right)$ factors which doubles the number of components, and
- the number of data association hypotheses $\left|\mathcal{D}(\mathcal{G})\right|$ for each graph.

Since processing of the full Gaussian mixture is infeasible, due to memory and processing limitations, approximations are necessary. For the data association problem, a common algorithm is the Joint Probabilistic Data Association (JPDA) filter [4]. The approach of the filter is to, at each time step, approximate a Gaussian mixture with a single Gaussian, using moment matching. We here describe how that algorithm can be extended to the case of resolution limitations. For the extension, two alternatives are proposed: either to calculate the full Gaussian mixture, and to approximate that with a

single Gaussian, or to perform a two-step approximation, which is less computationally intensive. The two approaches are described in the following two sections.

7.1 JPDA approximation of the full Gaussian mixture

The most accurate moment-matching approximation is given by calculating the full Gaussian mixture in (95), and then approximate that with a single Gaussian. A pseudo-code description of such a procedure is found in Table 1, and in the following we discuss the different steps in more detail. The algorithmic description starts with the predicted density $p\big(\mathbf{x}_k\big|\mathbf{Z}^{k-1}\big)$.

Step I: Generate all feasible graphs

The first step is to generate all feasible graphs, $\mathcal{G}$, which is done by first forming the set of possible resolution events, $\mathcal{R}$ (cf., for example, Example 1), and then generating the feasible resolution graphs for each event.

Step II: Data association hypotheses formulation

For each generated resolution graph, the set of data association hypotheses, $\mathcal{D}(\mathcal{G})$ is formulated. The set of data association hypotheses includes all combinations of group target-to-measurement assignments[1], including missed detections.

Step III: Measurement and resolution model update

The third step is divided into three sub-steps. All operations in step III are performed for each data association hypothesis formulated in Step II.

In Step III-a), the data association hypothesis probability is calculated according to (30).

Step III-b) is to update the predicted density function $\mathcal{N}\big(\mathbf{x}_k;\hat{\mathbf{x}}_{k|k},\mathbf{P}_{k|k}\big)$ with the measurement likelihood under the current resolution and data association hypotheses. The update can either be performed sequentially, as described in (41) and (42), or in a single calculation, given by (51). After the measurement update, we obtain the scaled Gaussian density $c^{\mathcal{G},\mathbf{d}}\mathcal{N}\big(\mathbf{x}_k;\hat{\mathbf{x}}^{\mathcal{G},\mathbf{d}}_{k|k},\mathbf{P}^{\mathcal{G},\mathbf{d}}_{k|k}\big)$ with proportionality weight (using (51))

$$c^{\mathcal{G},\mathbf{d}} = \Pr\big\{\mathbf{d}\big|\mathcal{G},\mathbf{x}_k\big\}\mathcal{N}\big(\mathbf{Z}_k;\breve{\mathbf{H}}\hat{\mathbf{x}}_{k|k-1},\mathbf{S}^{\mathcal{G},\mathbf{d}}\big). \tag{92}$$

[1]Note that several graphs lead to the same data association hypothesis. Thus, in the measurement update, only a single update is needed for those graphs. The resolution update with the graph probabilities, however, must be performed for each graph.

Table 1: Complete Gaussian mixture JPDA filter with resolution model

---

I: Generate all feasible graphs, $\mathcal{G}$.

II : For each graph, formulate all data association hypotheses, $\mathcal{D}(\mathcal{G})$.

III: For each data association hypothesis, $\mathbf{d} \in \mathcal{D}(\mathcal{G})$:

III-a) Calculate $\Pr\{\mathbf{d}|\mathcal{G}, \mathbf{x}_k\}$ according to (30).

III-b) Perform measurement update according to (41) and (42), or by (51). This yields a scaled Gaussian $c^{\mathcal{G},\mathbf{d}}\mathcal{N}\big(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^{\mathcal{G},\mathbf{d}}, \mathbf{P}_{k|k}^{\mathcal{G},\mathbf{d}}\big)$ with weight given by (92).

III-c) Update with resolution model according to (32), (34), and either (64)–(66), or by (86) and (87). The result is a sum of scaled Gaussians $\sum_{u\in\mathcal{U}(\mathcal{G})} c^{\mathcal{G},u,\mathbf{d}}\mathcal{N}\big(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^{\mathcal{G},u,\mathbf{d}}, \mathbf{P}^{\mathcal{G},u,\mathbf{d}}\big)$ with weights given by (93) .

IV: Approximate the Gaussian mixture in (95) using moment matching, according to (94), and (96)–(98).

In Step III-c), the resolution update is performed for each data association hypothesis. The general update is described in (32). As seen in (34), the resolution update results in a sum of products. For each summation term, the calculation can either be performed sequentially, as given by (64), (65) and (66), or in a single step, described by (86) and (87). After the update, we obtain a sum of scaled Gaussians $\sum_{u\in\mathcal{U}(\mathcal{G})} c^{\mathcal{G},u,\mathbf{d}}\mathcal{N}\big(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^{\mathcal{G},u,\mathbf{d}}, \mathbf{P}^{\mathcal{G},u,\mathbf{d}}\big)$, where the weight is given by

$$\begin{aligned} c^{\mathcal{G},u,\mathbf{d}} = \Pr\{\mathbf{d}|\mathcal{G}, \mathbf{x}_k\}\mathcal{N}\big(\mathbf{Z}_k; \breve{\mathbf{H}}\hat{\mathbf{x}}_{k|k-1}, \mathbf{S}^{\mathcal{G},\mathbf{d}}\big) \\ \times \big|2\pi\mathbf{I}_N \otimes \mathbf{R}_{u,N_{\text{res}}}\big|\mathcal{N}\big(\mathbf{0}; (\mathbf{D}^u \otimes \mathbf{I}_{N_{\text{res}}})\tilde{\mathbf{H}}\hat{\mathbf{x}}_{k|k}, \mathbf{S}^u\big). \end{aligned} \tag{93}$$

In (93), $\mathbf{D}^u$ represent the current resolution event (cf. (74) and (76)).

Step IV: Moment matching

The final step of the full Gaussian mixture JPDA algorithm with resolution modeling is the moment matching approximation. First, the weight components are normalized,

$$\bar{c}^{\mathcal{G},u,\mathbf{d}} = \frac{c^{\mathcal{G},u,\mathbf{d}}}{\sum_{\mathcal{G}}\sum_{u\in\mathcal{U}(\mathcal{G})}\sum_{\mathbf{d}\in\mathcal{D}(\mathcal{G})} c^{\mathcal{G},u,\mathbf{d}}}. \tag{94}$$

If the prior density is Gaussian, the end result after resolution and measurement updates is a density mixture of the form

$$p\big(\mathbf{x}_k\big|\mathbf{Z}^k\big) = \sum_{\mathcal{G}} \sum_{u\in\mathcal{U}(\mathcal{G})} \sum_{\mathbf{d}\in\mathcal{D}(\mathcal{G})} \bar{c}^{\mathcal{G},u,\mathbf{d}} \mathcal{N}\big(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^{\mathcal{G},u,\mathbf{d}}, \mathbf{P}_{k|k}^{\mathcal{G},u,\mathbf{d}}\big). \tag{95}$$

The final step of the first version of JPDA algorithm is to approximate this Gaussian mixture density with a single Gaussian. The best way of making that approximation, in the Kullback-Leibler sense, is to perform second-order moment matching. We thus make the approximation

$$p\big(\mathbf{x}_k\big|\mathbf{Z}^k\big) \approx \mathcal{N}\big(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^{A_1}, \mathbf{P}_{k|k}^{A_1}\big), \tag{96}$$

where

$$\hat{\mathbf{x}}_{k|k}^{A_1} = \sum_{\mathcal{G}} \sum_{u\in\mathcal{U}(\mathcal{G})} \sum_{\mathbf{d}\in\mathcal{D}(\mathcal{G})} \bar{c}^{\mathcal{G},u,\mathbf{d}} \hat{\mathbf{x}}_{k|k}^{\mathcal{G},u,\mathbf{d}} \tag{97}$$

$$\mathbf{P}_{k|k}^{A_1} = \sum_{\mathcal{G}} \sum_{u\in\mathcal{U}(\mathcal{G})} \sum_{\mathbf{d}\in\mathcal{D}(\mathcal{G})} \bar{c}^{\mathcal{G},u,\mathbf{d}} \Big\{ \mathbf{P}_{k|k}^{\mathcal{G},u,\mathbf{d}} + \big(\hat{\mathbf{x}}_{k|k}^{\mathcal{G},u,\mathbf{d}} - \hat{\mathbf{x}}_{k|k}^{A_1}\big)\big(\hat{\mathbf{x}}_{k|k}^{\mathcal{G},u,\mathbf{d}} - \hat{\mathbf{x}}_{k|k}^{A_1}\big)^T \Big\}. \tag{98}$$

7.2 Two-step JPDA approximation

When the number of targets is large and the clutter level high, the number of Gaussian components to calculate in each iteration can be large. Then, a less computationally demanding method is advantageous. We here present such a method, where instead of approximating the full Gaussian mixture, we perform a two-step procedure including two Gaussian approximations.

In Table 2, the two-step JPDA filter is described. In the following, we describe some of the steps in more detail. Since Step I to III-b) are the same as for the algorithm in Table 1, we start with Step IV.

Step IV: Moment matching over data association hypotheses

After the measurement update step for graph $\mathcal{G}$ and data association hypothesis $\mathbf{d}$, we obtain a scaled Gaussian $c^{\mathcal{G},\mathbf{d}}\mathcal{N}\big(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^{\mathcal{G},\mathbf{d}}, \mathbf{P}_{k|k}^{\mathcal{G},\mathbf{d}}\big)$ with weight given by (92). After having performed measurement updates for all data association hypotheses, we have a Gaussian mixture. Step IV of the two-step JPDA filter is to approximate this Gaussian mixture with a single Gaussian, using

moment matching[2], i.e.,

$$\sum_{\mathbf{d}\in\mathcal{D}(\mathcal{G})} \bar{c}^{\mathcal{G},\mathbf{d}} \mathcal{N}\big(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^{\mathcal{G},\mathbf{d}}, \mathbf{P}_{k|k}^{\mathcal{G},\mathbf{d}}\big) \approx \mathcal{N}\big(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^{\mathcal{G}}, \mathbf{P}_{k|k}^{\mathcal{G}}\big), \tag{99}$$

where

$$\bar{c}^{\mathcal{G},\mathbf{d}} = \frac{c^{\mathcal{G},\mathbf{d}}}{\sum_{\mathbf{d}\in\mathcal{D}(\mathcal{G})} c^{\mathcal{G},\mathbf{d}}} \tag{100}$$

$$\hat{\mathbf{x}}_{k|k}^{\mathcal{G}} = \sum_{\mathbf{d}\in\mathcal{D}(\mathcal{G})} \bar{c}^{\mathcal{G},\mathbf{d}} \hat{\mathbf{x}}_{k|k}^{\mathcal{G},\mathbf{d}} \tag{101}$$

$$\hat{\mathbf{P}}_{k|k}^{\mathcal{G}} = \sum_{\mathbf{d}\in\mathcal{D}(\mathcal{G})} \bar{c}^{\mathcal{G},\mathbf{d}} \Big\{ \hat{\mathbf{P}}_{k|k}^{\mathcal{G}} + \big(\hat{\mathbf{x}}_{k|k}^{\mathcal{G},\mathbf{d}} - \hat{\mathbf{x}}_{k|k}^{\mathcal{G}}\big)\big(\hat{\mathbf{x}}_{k|k}^{\mathcal{G},\mathbf{d}} - \hat{\mathbf{x}}_{k|k}^{\mathcal{G}}\big)^T \Big\}. \tag{102}$$

Step V: resolution update for each graph

After the moment matching in Step IV, we have a single Gaussian for each graph. In Step V, the resolution model update is performed under each graph. This gives a Gaussian mixture

$$\sum_{u\in\mathcal{U}(\mathcal{G})} \sum_{\mathbf{d}\in\mathcal{D}(\mathcal{G})} \bar{c}^{\mathcal{G},u} \mathcal{N}\big(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^{\mathcal{G},u}, \mathbf{P}_{k|k}^{\mathcal{V},u}\big),$$

with weights

$$\bar{c}^{\mathcal{G},u} = \frac{c^{\mathcal{G},u}}{\sum_{u\in\mathcal{U}(\mathcal{G})} \sum_{\mathbf{d}\in\mathcal{D}(\mathcal{G})} c^{\mathcal{G},u}}, \tag{103}$$

$$c^{\mathcal{G},u} = \big|2\pi\mathbf{R}_{u,N_{\text{res}}}\big| \mathcal{N}\big(\mathbf{0}; (\mathbf{D}^u \otimes \mathbf{I}_{N_{\text{res}}})\tilde{\mathbf{H}}\hat{\mathbf{x}}_{k|k}^{\mathcal{G},u}, \mathbf{S}^{\mathcal{G},u}\big), \tag{104}$$

where $\mathbf{S}^{\mathcal{G},u}$ is given by (88), inserting $\mathbf{P}_{k|k}^{\mathcal{G}}$ instead of $\mathbf{P}_{k|k}$.

Step VI: Gaussian approximation of posterior density

For each graph, $\mathcal{G}$, Step V provides a Gaussian mixture. The posterior density is hence approximated by

$$p\big(\mathbf{x}_k\big|\mathbf{Z}^k\big) \approx \sum_{\mathcal{G}} \sum_{u\in\mathcal{U}(\mathcal{G})} \bar{c}^{\mathcal{G},u} \mathcal{N}\big(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^{\mathcal{G},u}, \mathbf{P}_{k|k}^{\mathcal{G},u}\big). \tag{105}$$

[2]In fact, this the standard JPDA approximation performed for each resolution graph.

Table 2: Two-step JPDA filter with resolution model

I: Generate all feasible graphs, $\mathcal{G}$.

II : For each graph, formulate all data association hypotheses, $\mathcal{D}(\mathcal{G})$

III: For each data association hypothesis, $\mathbf{d} \in \mathcal{D}(\mathcal{G})$

III-a) Calculate $\Pr\left\{\mathbf{d}\middle|\mathcal{G}, \mathbf{x}_k\right\}$ according to (30)

III-b) Update the predicted density with information from measurements, according to (41) and (42), or by (51). This yields a scaled Gaussian $c^{\mathcal{G},\mathbf{d}}\mathcal{N}\big(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^{\mathcal{G},\mathbf{d}}, \mathbf{P}_{k|k}^{\mathcal{G},\mathbf{d}}\big)$ with weight given by (92).

IV: Perform moment matching over the data association hypotheses, as described in (99)–(102). This gives a single Gaussian $\mathcal{N}\big(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^{\mathcal{G}}, \mathbf{P}_{k|k}^{\mathcal{G}}\big)$.

V: For each graph, perform an update with the resolution model. This yields a Gaussian mixture, $\sum_{u\in\mathcal{U}(\mathcal{G})}\sum_{\mathbf{d}\in\mathcal{D}(\mathcal{G})} c^{\mathcal{G},u}\mathcal{N}\big(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^{\mathcal{G},u}, \mathbf{P}_{k|k}^{\mathcal{G},u}\big)$, with weights given by (103), and mean value and covariance matrix by (101) and (102).

VI: Approximate the Gaussian mixture in (105) with a single Gaussian, as described in (106)–(108).

The final step of the algorithm is to make a second moment matching, to approximate the posterior density as a single Gaussian, according to

$$p\big(\mathbf{x}_k\big|\mathbf{Z}^k\big) \approx \mathcal{N}\big(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^{A_2}, \mathbf{P}_{k|k}^{A_2}\big), \tag{106}$$

where

$$\hat{\mathbf{x}}_{k|k}^{A_2} = \sum_{\mathcal{G}} \sum_{u\in\mathcal{U}(\mathcal{G})} \bar{c}^{\mathcal{G},u} \hat{\mathbf{x}}_{k|k}^{\mathcal{G},u} \tag{107}$$

$$\mathbf{P}_{k|k}^{A_2} = \sum_{\mathcal{G}} \sum_{u\in\mathcal{U}(\mathcal{G})} \bar{c}^{\mathcal{G},u} \Big\{ \mathbf{P}_{k|k}^{\mathcal{G},u} + \big(\hat{\mathbf{x}}_{k|k}^{\mathcal{G},u} - \hat{\mathbf{x}}_{k|k}^{A_2}\big)\big(\hat{\mathbf{x}}_{k|k}^{\mathcal{G},u} - \hat{\mathbf{x}}_{k|k}^{A_2}\big)^T \Big\}. \tag{108}$$

8 Simulations

In this section, we evaluate the JPDA filter with the proposed resolution model, and compare the results to those of the JPDA filter without a resolution model. The considered tracking scenario includes three targets whose trajectories are shown in Fig. 4. The targets move with a constant speed of 5 m/s, and their separation is 60 m in the middle part of the scenario.

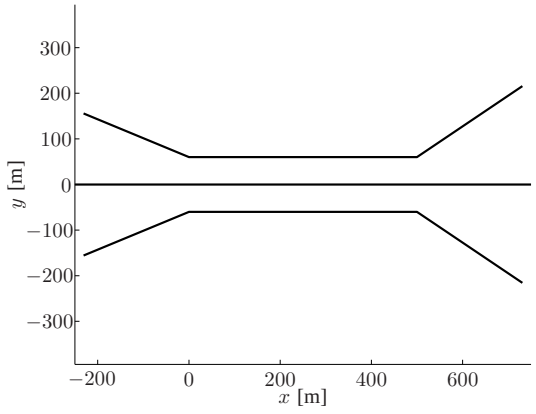

Figure 4: Trajectories for the three-target scenario. The separation between the targets is 60 m in the middle, and the targets move from left to right.

As state vector, we use $\mathbf{x} = \begin{bmatrix} x & y & \dot{x} & \dot{y} \end{bmatrix}$. For prediction of future states in the filters, a (nearly) constant velocity model is used, with system matrix

(cf. (4)) $\mathbf{F} = \text{diag}\{\tilde{\mathbf{F}}, \tilde{\mathbf{F}}, \tilde{\mathbf{F}}\}$, where

$$\tilde{\mathbf{F}} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{109}$$

and where T is the time between measurements. Further, the process noise covariance matrix is $\mathbf{Q} = \text{diag}\{\tilde{\mathbf{Q}}, \tilde{\mathbf{Q}}, \tilde{\mathbf{Q}}\}$, where

$$\tilde{\mathbf{Q}} = q_0 \begin{bmatrix} T^3/3 & 0 & T^2/2 & 0 \\ 0 & T^3/3 & 0 & T^2/2 \\ T^2/2 & 0 & T & 0 \\ 0 & T^2/2 & 0 & T \end{bmatrix}, \tag{110}$$

and $q_0 = 0.2$, which is the parameter value that gives the best average performance for this scenario.

At regular time intervals, T, of one second, a sensor, which is located at $(-10000, -10000)$, provides measurements of the range R and azimuth angle φ to the targets. The accuracy of the measurements depends on whether the targets are resolved or not, as discussed in Section 4. More specifically, the range errors are assumed to be constant as the number of targets in an unresolved group increases, while the azimuth errors are assumed to increase due to radar target glint. The target-number dependent measurement noise covariance matrix is thus

$$\mathbf{R}^{n_g} = \begin{bmatrix} \left(\sigma_R\right)^2 & 0 \\ 0 & \left(\sigma_\varphi^{n_g}\right)^2 \end{bmatrix}, \tag{111}$$

where $\sigma_\varphi^{n_g} = n_g \cdot 0.1\pi/180$. Further, the resolution capability of the radar sensor is given by the parameters (cf. (19)) α_R equal to 60 m and α_φ equal to $0.3\pi/180$. With these parameters, averaged over 100 trials for the 200 time steps of the scenario, all three targets are resolved 79.7 times (time steps), two targets are unresolved 70.7 times and all three targets form a group target 49.6 times.

The detection probability P_D of the sensor is assumed equal for all group sizes. In the evaluations, we test detection probabilities of 0.999 and 0.95. On top of target-generated measurements, the sensor also reports spurious measurements due to clutter. We evaluate the results for two different clutter levels: low clutter, with 1 false measurement per scan on average (and rarely more than 4), and moderate clutter, with 4 false measurements per scan on average (and rarely more than 8).

In Figs. 5 and 6, examples of trajectory estimates from the JPDA filter with and without the resolution model are shown, for the case of no clutter and a detection probability of 0.999. By comparing the figures, it is seen that the filter which takes resolution limitations into account produces more stable, and well-separated, tracks.

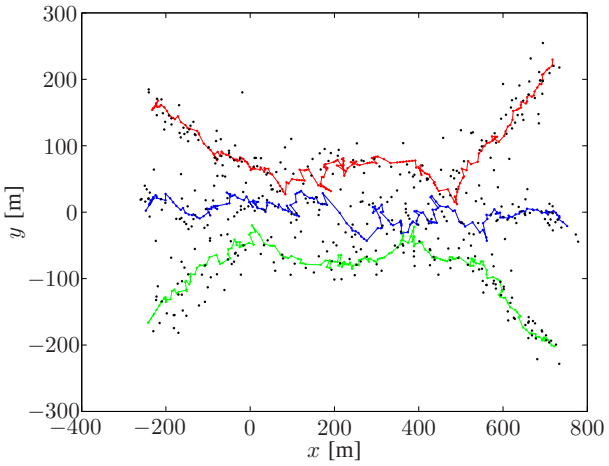

Figure 5: Example of the output of the standard JPDA filter for the case of no clutter and a detection probability of 0.999, but with resolution limitations expressed by $\alpha_R = 60$ m and $\alpha_\varphi = 0.3°$. The dots represent the sensor measurements converted to Cartesian coordinates.

As a measure of performance, we use the Mean Optimal Subpattern Assignment (MOSPA) measure [19]. The measure does not consider target identity, but finds the best permutation of the targets at each time instant, which minimizes the sum of distances between estimate and truth. Since our interest lies in describing data, and since the radar data comes without labeling, the MOSPA is a natural performance measure to choose. As basis measure, we use the Euclidean distance. We further use the first-order MOSPA measure ($p = 1$, see [19]), which means that the unit of the MOSPA distance is meters. As cut-off value we use $c = 300$ m.

For the performance evaluations, we first consider the case of a detection probability of 0.999. In Fig. 7, the MOSPA performance over 100 Monte Carlo runs is presented for the low-clutter case for the JPDA filter without resolution model, and for the full Gaussian mixture JPDA filter with a resolution model. As a reference, the performance of the JPDA filter when the sensor has perfect resolution is also shown. As seen, when the targets are closely spaced and the resolution limitations have effect, the JPDA filter without resolution model performs worse than the JPDA filter with a

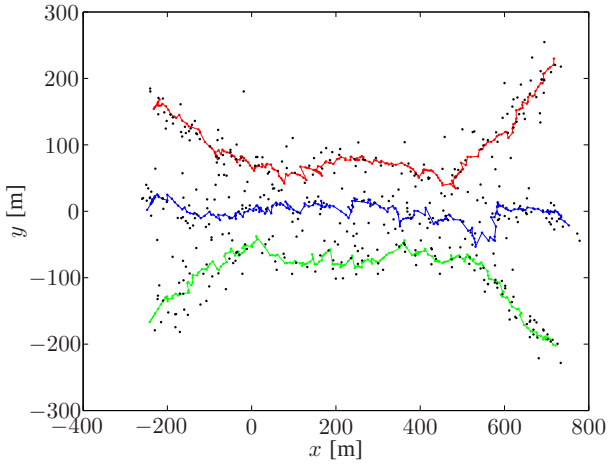

Figure 6: Example of the output of the full Gaussian mixture JPDA filter with a resolution model, for the case of no clutter and a detection probability of 0.999, but with resolution limitations expressed by $\alpha_R = 60$ m and $\alpha_\varphi = 0.3°$. The dots represent the sensor measurements converted to Cartesian coordinates.

resolution model, which provides evidence that the resolution model indeed accounts for a better description of the data.

In Fig. 8, the MOSPA results for the moderate clutter level are shown. There, it is seen that the JPDA filter performs significantly worse than with the lower clutter level, whereas the performance of the JPDA filter with a resolution model, performs almost as well. The reason for the significantly worse performance of the JPDA filter is that a large number of track losses occur (where tracks are attracted to clutter detections, and move away from the true trajectory). For the JPDA filter with a resolution model, on the other hand, the attraction of the clutter detections is weaker due to the filter considering the events that the targets are not resolved.

The final evaluation regards a lower detection probability of 0.95, and a moderate clutter level. The results for this setup are shown in Fig. 9. By comparing this figure with the results shown in Fig. 8, it is seen that the performance of the JPDA filter is somewhat better with the lower detection probability than with the higher one. The reason for this is that the weights of the data association hypotheses under which clutter detections are assigned to the tracks are lower when the detection probability is lower, and we thus obtain more stable tracks. Still, the performance is significantly improved by using the proposed sensor resolution model.

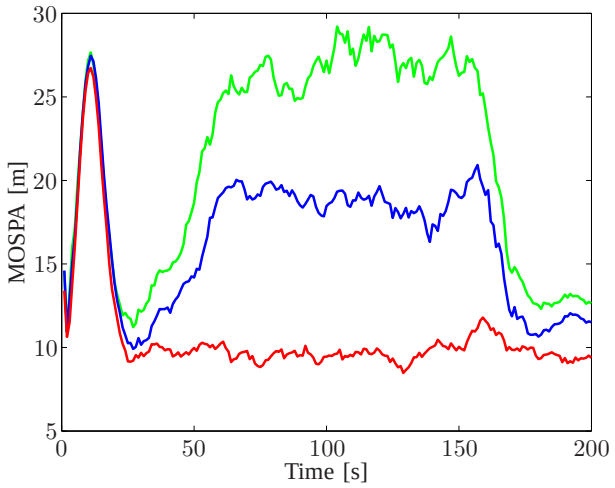

Figure 7: MOSPA performance for the JPDA filters averaged over 100 Monte-Carlo runs for a detection probability of 0.999 and one clutter measurement on average. Green: without resolution model, blue: with a resolution model, and red: with perfect resolution.

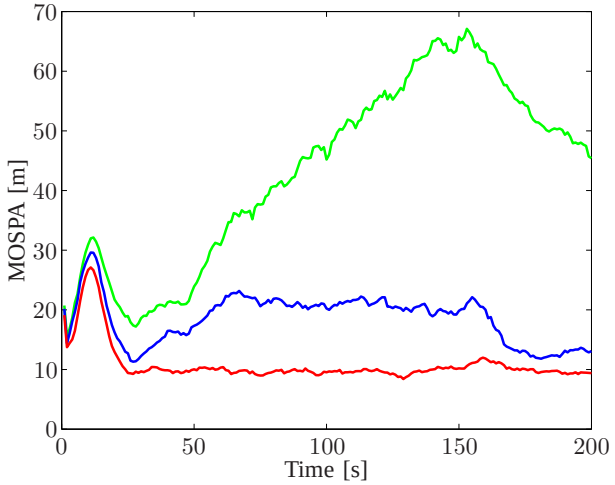

Figure 8: MOSPA performance for the JPDA filters averaged over 100 Monte-Carlo runs for a detection probability of 0.999 and four clutter measurements on average. Green: without resolution model, blue: with a resolution model, and red: with perfect resolution.

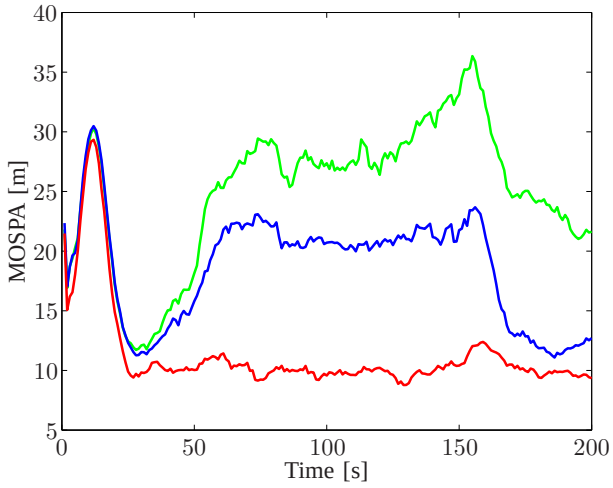

Figure 9: MOSPA performance for the JPDA filters averaged over 100 Monte-Carlo runs for a detection probability of 0.95 and four clutter measurement on average. Green: without resolution model, blue: with a resolution model, and red: with perfect resolution.

9 Conclusion

In this article, we consider the modeling of limited sensor resolution for an arbitrary, but known, number of targets. The main contribution is a framework for handling resolution limitations, which can be easily incorporated into traditional Bayesian tracking filters. The framework relies on a graph description of a resolution event, and on modeling the resolution probability as independent between target pairs. To complete the framework and to attain a multitarget resolution model, a model for the resolution probability for two targets and a group measurement model for an arbitrary number of targets are needed. By suggesting two such models, the exact calculation of the posterior probability density function under both data association and resolution conflicts is described. Under linear and Gaussian assumptions, the posterior density is a Gaussian mixture, and the components of that mixture are also derived in the paper. Further, two alternative approximations of the density mixture by a single Gaussian are proposed, which both can be considered as extensions of the Joint Probabilistic Data Association (JPDA) filter taking resolution problems into account. Finally, the JPDA filters with and without a resolution model are evaluated on a three-target tracking scenario with simulated radar data. The results show significantly improved tracking performance of the resolution filters for all considered setups.

References

[1] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood, MA: Artech House, 1999.

[2] F. Daum and R. Fitzgerald, "Importance of resolution in multiple-target tracking," in *Signal and Data Processing of Small Targets*, ser. Proc. SPIE, vol. 2235, 1994.

[3] K. Chang and Y. Bar-Shalom, "Joint probabilistic data association with possibly unresolved measurements and maneuvers," *IEEE Transactions on Automatic Control*, vol. 29, no. 7, pp. 585–594, July 1984.

[4] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE Journal of Oceanic Engineering*, vol. 8, no. 3, pp. 173–183, July 1983.

[5] S. Mori, K. Chang, and C. Chong, "Tracking aircraft by acoustic sensors—multiple hypothesis approach applied to possibly unresolved measurements," in *American Control Conference*, June 1987, pp. 1099–1105.

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

[7] S. Deb, M. Yeddanapudi, K. Pattipati, and Y. Bar-Shalom, "A generalized s-d assignment algorithm for multisensor-multitarget state estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 2, pp. 523–538, April 1997.

[8] S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, January 2004.

[9] W. Koch and G. van Keuk, "Multiple hypothesis track maintenance with possibly unresolved measurements," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 3, pp. 883–892, July 1997.

[10] M. Waxman and O. Drummond, "A bibliography of cluster (group) tracking," in *Signal and Data Processing of Small Targets*, ser. Proc. SPIE, vol. 5428, June 2004.

[11] D. Svensson, M. Ulmke, and L. Danielsson, "Multitarget sensor resolution model for arbitrary target numbers," in *Signal and Data Processing of Small Targets*, ser. Proc. SPIE, vol. 7698, 2010.

[12] ——, "Joint probabilistic data association filter for partially unresolved target groups," in *Proceedings of the 13th International Conference on Information Fusion*, July 2010.

[13] K. Chang and Y. Bar-Shalom, "A simplification of the JPDAM algorithm," *IEEE Transactions on Automatic Control*, vol. 31, no. 10, pp. 989–981, October 1986.

[14] J. Gunnarsson, L. Svensson, L. Danielsson, and F. Bengtsson, "Tracking vehicles using radar detections," in *IEEE Intelligent Vehicles Symposium*, 2007, pp. 296–302.

[15] R. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82 (Series D), pp. 35–45, 1960.

[16] A. Jazwinski, *Stochastic Processes and Filtering Theory*. New York: Academic, 1970, ch. 8, pp. 272–281.

[17] S. Julier and J. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, March 2004.

[18] J. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Transactions on Circuits and Systems*, vol. CAS-25, no. 9, pp. 772–781, September 1978.

[19] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, August 2008.