Mark Ryan Leonard

Robust Signal Processing in Distributed Sensor Networks

Dissertation

TECHNISCHE UNIVERSITÄT DARMSTADT

Robust Signal Processing in Distributed Sensor Networks

Dem Fachbereich
Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
vorgelegte Dissertation
von

Mark Ryan Leonard, M.Sc.

Erstgutachter: Prof. Dr.-Ing. Abdelhak M. Zoubir
Zweitgutachter: Prof. Dr. Sergio Barbarossa

Darmstadt 2019

Leonard, Mark Ryan — Robust Signal Processing in Distributed Sensor Networks
Darmstadt, Technische Universität Darmstadt,
Jahr der Veröffentlichung der Dissertation auf TUprints: 2019
URN: urn:nbn:de:tuda-tuprints-84896
Tag der mündlichen Prüfung: 15. Februar 2019

"Start every week with a break-neck urgent design
And end every speed day with my briefcase representing free time
Spending my fruit as my purchases become my lifeline
Please give my love to my family
I'll doubtfully be home at christmas time

Don't disturb me in this state
Please leave me purgatorying
I'll be damned if I'm to wake
This is far more than I am equipped for."

Alanis Morissette – Purgatorying

To Bennett

"Flowers are more beautiful than they are thirty-eight."

Oren Lavie – The Bear Who Wasn't There

Acknowledgments

I wish to thank everyone, who contributed to this thesis in various ways and supported me during this truly transformative time of my life.

First of all, I would like to thank Prof. Dr.-Ing. Abdelhak M. Zoubir for giving me the opportunity to pursue my Ph.D. at the Signal Processing Group. Thank you for your continued guidance and support, your understanding, and for all the freedom that allowed me to persevere and finish this project in the first place.

Thanks to Prof. Dr. Sergio Barbarossa for being my co-advisor. I really appreciated your valuable comments on my work. I also wish to thank Prof. Dr.-Ing. Rolf Jakoby and Prof. Dr.-Ing. Anja Klein for making my defense as pleasurable as an exam can be.

I sincerely want to thank Renate Koschella for everything you have done and continue to do for me and the entire group. Thank you for your compassion and support and the countless encouraging, calming, and liberating talks over the years. I am certain that a lot of doctoral candidacies would have ended differently if it weren't for you.

Big thanks go to Dr.-Ing. Michael Fauß. Thank you for your guidance and for always helping me whenever I was stuck. Also, thanks for proof-reading virtually all of my publications—including this thesis—and for finding even the most unspottable errors.

I also wish to thank my various roommates over the years, starting with Dr.-Ing. Stefan Leier. Thank you for your clear words although I wouldn't listen. Thank you Dr.-Ing. Nevine Demitri for helping me turn our office into the best coffee shop in town. Thank you to Di Jin for transforming it into the best tea house in the western hemisphere. Dominik Reinhard, thank you for all the sarcastic comments, which made even bad times more endurable. Also thanks for helping me in keeping the office well-ventilated at all times.

Thank you to my partner in crime Freweyni Kidane Teklehaymanot for finishing this together. Also, thanks for the wonderful times in Calgary and Rome. I'll never forget the great vegan food we had and how we discovered the entryway to the underworld. And you should always remember the secret ;)

Thanks to Dr.-Ing. Lala Khadidja Hamaidi. Merci beaucoup de me tenir compagnie dans la contrée sauvage.

Thank you Dr.-Ing. Sahar Khawatmi for your cheerful nature and all the fun we had trying to figure out German grammar.

Thanks to Hauke Fath for always trying your best in making our system work.

I would like to thank all of my other current and former colleagues and members of the group, including Dr.-Ing. Sara Al-Sayed, Dr.-Ing. Mouhammad Alhumaidi, Patricia Binder, Jack Dagdagan, Dr.-Ing. Christian Debes, Dr.-Ing. Tai Fei, Dr. Stefano Fortunati, Dr.-Ing. Gök-

han Gül, Dr.-Ing. Jürgen Hahn, Prof. Dr.-Ing. Eberhard Hänsler, Dr.-Ing. Philipp Heidenreich, Dr. Roy Howard, Huiping Huang, Amare Kassaw, Dr.-Ing. Michael Leigsnering, Toufik Mouchini, Dr.-Ing. Michael Muma, Afief Dias Pambudi, Prof. Dr.-Ing. Henning Puder, Ziliang Qiao, Dr.-Ing. Simon Rosenkranz, Dr.-Ing. Tim Schäck, Ann-Kathrin Seifert, Dr.-Ing. Adrian Šošić, Sergey Sukhanov, Dr.-Ing. Wassim Suleiman, Dr.-Ing. Fiky Suratman, Dr.-Ing. Christian Weiss, Dr.-Ing. Feng Yin, and Dr.-Ing. Wenjun Zeng. You made these last five years very enjoyable.

Thank you Roman Hocke for giving me the opportunity to pursue my other passion during these past five years—and for half of my life. Thanks to Lisa Blenninger, Claudia von Hornstein, Markus Michalek and Dr. Cornelia Petersen-Laux for the great collaboration over all these years.

Special thanks go to Sabine Schmelzer for fixing me up and to Salima Elyazidi for always finding time for that.

Finally I would like to thank my family. Thanks to my parents Gerhard and Renate, my sister Jennifer, and my mother-in-law Monika for your unconditional love and support and for always believing in me. Thank you Jonathan, Tineke, and Hugo for providing the best and most relaxing seaside getaways one could wish for. Thanks to Fee, Lotta, Lotti, Mäxchen, Merle, Samson, and Sissi for all the joy and consolation.

Thank you René for these intense last three months, a wonderful Christmas eve, and for holding out until after my defense. You will be missed very much.

Thank you to my wife Fabienne. Thank you for all that you are, all that you made me, and all that we will become.

Last but not least, thank you Bennett for making clear, what's next.

Mark Ryan Leonard 04. Dezember 2018

Robuste Signalverarbeitung in verteilten Sensornetzen

Kurzfassung

Statistische Robustheit und kollaborative Inferenz in einem verteilten Sensornetz sind zwei anspruchsvolle Herausforderungen, die an viele moderne Signalverarbeitungsanwendungen gestellt werden. Diese Dissertation hat zum Ziel, gemeinsame Lösungen für diese Aufgaben zu entwickeln und generische Algorithmen bereitzustellen, die auf eine Vielzahl von Problemen aus der Praxis anwendbar sind.

Der erste Teil der Arbeit befasst sich mit der sequentiellen Detektion – einem Teilgebiet der Detektionstheorie, das sich auf die Entscheidungsfindung basierend auf einer möglichst geringen Anzahl von Messwerten konzentriert. Nach der Betrachtung einiger grundlegender Konzepte des statistischen Hypothesentests wird eine allgemeine Formulierung des sequentiellen Konsens+Innovationen Likelihood-Quotienten-Tests (Consensus+Innovations Sequential Probability Ratio Tests) zum sequentiellen Testen binärer Hypothesen in verteilten Netzen hergeleitet. In einem nächsten Schritt werden mehrere robuste Versionen des Algorithmus' entwickelt, die auf zwei verschiedenen Robustheitsparadigmen basieren. Die Funktionalität der vorgestellten Detektoren wird in Simulationen verifiziert und ihre Leistung wird unter verschiedenen Netzwerkbedingungen sowie Ausreißerkonzentrationen untersucht. Anschließend wird das Konzept auf mehrere Hypothesen ausgedehnt, indem es mit dem sequentiellen Matrix-Likelihood-Quotienten-Test (Matrix Sequential Probability Ratio Test) fusioniert wird. Ferner werden robuste Versionen des resultierenden Algorithmus' entwickelt. Die Leistungsfähigkeit der vorgeschlagenen Algorithmen wird in Simulationen verifiziert und evaluiert. Schließlich wird die Dempster-Shafer Evidenztheorie erstmals in der Literatur auf verteilte sequentielle Hypothesentests angewendet. Nach der Vorstellung einer neuartigen Methode für die Zuweisung von Wahrscheinlichkeitsmassen (Basic Probability Assignment) wird ein evidenzbasierter sequentieller Detektor für den Einsatz in verteilten Sensornetzen entwickelt und seine Leistungsfähigkeit in Simulationen verifiziert.

Der zweite Teil der Dissertation beschäftigt sich mit der Mehrziel-Verfolgung in verteilten Sensornetzen. Das Problem der Datenzuordnung wird diskutiert und die betrachteten Zustandsraum- und Messmodelle werden eingeführt. Als Nächstes werden das Konzept der zufälligen endlichen Mengen sowie die Wahrscheinlichkeitshypothesendichte-Filterung (Probability Hypothesis Density Filtering) beleuchtet. Anschließend wird eine neuartige verteilte Partikel-Filter Implementierung des Wahrscheinlichkeitshypothesendichte-Filters entwickelt, die auf einem zweistufigen Kommunikationsschema basiert. Es wird sowohl eine robuste als

auch eine zentralisierte Version des Algorithmus' hergeleitet. Darüber hinaus werden Rechenkomplexität und Kommunikationslast des verteilten sowie des zentralen Verfolgungsalgorithmus' analysiert. Schließlich werden Simulationen durchgeführt, in denen die vorgeschlagenen Methoden mit einem existierenden verteilten Verfolgungsalgorithmus verglichen werden. Zu diesem Zweck wird eine verteilte Version der A-posteriori-Cramér-Rao-Schranke (Posterior Cramér-Rao Lower Bound) entwickelt, die als Leistungsgrenze dient. Die Ergebnisse zeigen, dass die vorgestellten Algorithmen unter verschiedenen Umgebungsbedingungen gut funktionieren und die Konkurrenz übertreffen.

Mark Ryan Leonard December 04, 2018

Robust Signal Processing in Distributed Sensor Networks

Abstract

Statistical robustness and collaborative inference in a distributed sensor network are two challenging requirements posed on many modern signal processing applications. This dissertation aims at solving these tasks jointly by providing generic algorithms that are applicable to a wide variety of real-world problems.

The first part of the thesis is concerned with sequential detection—a branch of detection theory that is focused on decision-making based on as few measurements as possible. After reviewing some fundamental concepts of statistical hypothesis testing, a general formulation of the Consensus+Innovations Sequential Probability Ratio Test for sequential binary hypothesis testing in distributed networks is derived. In a next step, multiple robust versions of the algorithm based on two different robustification paradigms are developed. The functionality of the proposed detectors is verified in simulations, and their performance is examined under different network conditions and outlier concentrations. Subsequently, the concept is extended to multiple hypotheses by fusing it with the Matrix Sequential Probability Ratio Test, and robust versions of the resulting algorithm are developed. The performance of the proposed algorithms is verified and evaluated in simulations. Finally, the Dempster-Shafer Theory of Evidence is applied to distributed sequential hypothesis testing for the first time in the literature. After introducing a novel way of performing the basic probability assignment, an evidence-based sequential detector for application in distributed sensor networks is developed and its performance is verified in simulations.

The second part of the thesis deals with multi-target tracking in distributed sensor networks. The problem of data association is discussed and the considered state-space and measurement models are introduced. Next, the concept of random finite sets as well as Probability Hypothesis Density filtering are reviewed. Subsequently, a novel distributed Particle Filter implementation of the Probability Hypothesis Density Filter is developed, which is based on a two-step communication scheme. A robust as well as a centralized version of the algorithm are derived. Furthermore, the computational complexity and communication load of the distributed as well as the centralized trackers are analyzed. Finally, simulations are performed to compare the proposed algorithms with an existing distributed tracker. To this end, a distributed version of the Posterior Cramér-Rao Lower Bound is developed, which serves as a performance bound. The results show that the proposed algorithms perform well under different environmental conditions and outperform the competition.

Publications

The following publications have been produced during the period of doctoral candidacy.

Internationally Refereed Journal Articles

M. R. Leonard & A. M. Zoubir. "Multi-Target Tracking in Distributed Sensor Networks using Particle PHD Filters." In: *Signal Processing* 159 (June 2019), pp. 130–146.

M. R. Leonard & A. M. Zoubir. "Robust Sequential Detection in Distributed Sensor Networks." In: *IEEE Transactions on Signal Processing* 66.21 (Nov. 2018), pp. 5648–5662.

Internationally Refereed Conference Papers

M. R. Leonard, M. Stiefel, M. Fauß & A. M. Zoubir. "Robustifying Sequential Multiple Hypothesis Tests in Distributed Sensor Networks." In: *Proceedings of the 26th European Signal Processing Conference (EUSIPCO)*. Sept. 2018.

M. R. Leonard, C. A. Schroth & A. M. Zoubir. "Dempster-Shafer Theory Based Robust Sequential Detection in Distributed Sensor Networks." In: *Proceedings of the IEEE Statistical Signal Processing Workshop (SSP)*. June 2018.

M. R. Leonard, M. Stiefel, M. Fauß & A. M. Zoubir. "Robust Sequential Testing of Multiple Hypotheses in Distributed Sensor Networks." In: *Proceedings of the 43nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Mar. 2018.

M. R. Leonard & A. M. Zoubir. "Robust Distributed Sequential Hypothesis Testing for Detecting a Random Signal in Non-Gaussian Noise." In: *Proceedings of the 25th European Signal Processing Conference (EUSIPCO)*. Sept. 2017.

W. Hou, M. R. Leonard & A. M. Zoubir. "Robust Distributed Sequential Detection via Robust Estimation." In: *Proceedings of the 25th European Signal Processing Conference (EUSIPCO)*. Sept. 2017.

J. Rambach, M. F. Huber, M. R. Balthasar & A. M. Zoubir. "Collaborative Multi-Camera Face Recognition and Tracking." In: *Proceedings of the 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. Aug. 2015.

International Conference Papers

M. R. Balthasar, S. Al-Sayed, S. Leier & A. M. Zoubir. "Optimal Area Coverage in Autonomous Sensor Networks." In: *Proceedings of the 2nd International Conference and Exhibition on Underwater Acoustics (UA2014)*. Invited Paper. June 2014.

Filed Patent Applications

J. Rambach, M. Huber & M. R. Balthasar. "Method Of Distributed Face Recognition And System Thereof." Patent Application WO 2017/125915 A1 (World Intellectual Property Organization). July 27, 2017.

J. Rambach, M. Huber & M. R. Balthasar. "Method Of Distributed Face Recognition And System Thereof." Patent Application US 2017/0206403 A1 (United States). July 20, 2017.

Contents

1 Introduction

An increasing number of modern signal processing applications have to prevail under two fundamental circumstances: On the one hand, non-Gaussian disturbances such as impulsive noise call for robust solutions that accomplish the balancing act between optimality under nominal conditions and reliability in the face of distributional uncertainties [Huber, 1964; Huber, 1965; Huber & Strassen, 1973; Huber, 1981; Levy, 2008; Zoubir et al., 2012; Gül & Zoubir, 2017a; Zoubir et al., 2018]. On the other hand, the growing tendency to network ever more capable sensors and devices demands fully distributed architectures that do away with error-prone and communication-intensive central processing units [Cattivelli & Sayed, 2011; Tu & Sayed, 2011; Sayed, 2013; Balthasar et al., 2014; Leonard & Zoubir, 2019; Matta et al., 2016]. The dissertation at hand is focused on jointly solving these problems for a wide variety of real-world applications—whether it be air and ground traffic control, climate or health monitoring, smart homes and cities, or video surveillance [Tartakovsky et al., 2014]. To this end, generic algorithms are derived that can be easily deployed in distributed network architectures. Furthermore, a balance is struck between approximating the performance of centralized solutions and providing resilience in the face of unknown disturbances, all the while accounting for resource constraints at the individual network agents.

The aforementioned real-world applications and many others are based on two consecutive steps that are elementary to statistical signal processing: the *detection* of a phenomenon, and the subsequent *estimation*—and possibly tracking—of relevant parameters. While, in the first step, it is imperative to register new phenomena and parameter shifts instantly, step two requires continuous, accurate estimates and predictions of the parameters of interest. A separate part is dedicated to each of these two steps.

1.1 Sequential Detection

The first part of the thesis is concerned with a particular branch of detection theory: sequential detection [Wald, 1945; Wald, 1947; Novikov, 2009a; Novikov, 2009b; Tartakovsky et al., 2014; Fauß & Zoubir, 2015; Fauß, 2016]. While originally introduced in the 1940s by Wald, sequential hypothesis testing has been gaining traction recently as a growing number of applications require accurate decision-making in a timely manner. The objective is to make a reliable decision for one out of two or more hypotheses based on as few measurements as possible, which can reduce the testing time by up to 50% on average. To this end, a test statistic is continually updated as new samples are taken, and the threshold comparisons are repeated until the gathered information warrants an accurate decision with respect to a specified confidence measure.

Robustness, sequentiality, and a distributed network architecture are challenging, partially contradictory requirements to pose on a statistical hypothesis test. While combinations of either two of them—i.e., distributed sequential detection [Teneketzis & Ho, 1987; Blum et al., 1997; Sahu & Kar, 2014; Sahu & Kar, 2016; Liu & Mei, 2017; Li & Wang, 2018], robust sequential detection [DeGroot, 1960; Schmitz, 1987; Fauß & Zoubir, 2016; Gül & Zoubir, 2017a], and robust hypothesis testing in distributed sensor networks [Veeravalli et al., 1994; Blum et al., 1997; Gül, 2017; Gül & Zoubir, 2017b; Al-Sayed et al., 2017]—have received considerable attention in recent years, their complete union has not been treated in the literature, yet.

1.2 Location Estimation & Tracking

The second part of the thesis considers the problem of estimating and tracking the location of multiple targets at once with the help of a distributed sensor network. This task is becoming increasingly relevant in many military and civilian applications including air and ground traffic control, harbor surveillance, maritime traffic control, and video communication and surveillance [Challa et al., 2011; Maresca et al., 2014; Rambach et al., 2015].

A distributed network architecture offers several properties that make it desirable for tracking applications in general [Olfati-Saber et al., 2007]. The state-of-the-art of distributed *single-target* tracking is well summarized in [Hlinka et al., 2013]. Distributed versions of the Kalman Filter [Hlinka et al., 2013; Cattivelli et al., 2008] and the Particle Filter [Arulampalam et al., 2002] suffer from the problem of data association and cannnot be applied directly to multi-target tracking. Methods like the Joint Probabilistic Data Association Filter [Bar-Shalom et al., 2011] or the Multiple Hypothesis Tracker [Reid, 1979] address this problem in the single-target case but the resource constraints arising in sensor networks pose a challenge on the development of distributed implementations thereof [Oh et al., 2007].

The algorithms considered in the dissertation at hand are based on the concept of Probability Hypothesis Density (PHD) filtering [Mahler, 2003; Clark, 2006], which circumvents the data association issue by modeling the tracking problem with the help of random finite sets. Furthermore, the focus is on sensor networks with maximum area coverage and neighborhood communication. Other solutions based on the PHD Filter have been studied, e.g., in [Uney et al., 2010; Uney et al., 2013; Battistelli et al., 2013]. Contrary to the methods presented in this work, these approaches either assume overlapping fields of view or employ a pairwise communication scheme. The common idea, however, is to extend the single-sensor PHD Filter to the multi-sensor case through communication between multiple nodes, or nodes and a fusion center.

1.3 Overview & Contributions

The dissertation is divided into two parts. Before diving into each of them, Chapter 2 gives an overview of the fundamental concepts used throughout the thesis—statistical robustness and distributed sensor networks.

Part I: Sequential Detection

The first part investigates robust sequential hypothesis testing in distributed sensor networks with the goal of obtaining a networkwide decision on the considered phenomenon. In Chapter 3, the basics of fixed-sample-size hypothesis testing, sequential hypothesis testing, and least favorable densities (LFDs) are reviewed.

Chapter 4 is concerned with sequential binary hypothesis tests in distributed sensor networks. Here, a general formulation of the Consensus+Innovations Sequential Probability Ratio Test ($\mathcal{CI}$SPRT)—originally introduced in [Sahu & Kar, 2014; Sahu & Kar, 2016]—is derived, which is not only applicable to arbitrary binary hypothesis tests but also suitable for real-world distributed detection problems. Furthermore, multiple robust versions of the $\mathcal{CI}$SPRT are developed based on two different robustification paradigms, namely, LFDs and robust estimators. Finally, the proposed algorithms are evaluated in simulated shift-in-mean and shift-in-variance tests.

The problem of sequentially testing multiple hypotheses in a distributed network architecture is the topic of Chapter 5. First, the Matrix Sequential Probability Ratio Test (MSPRT) for testing multiple hypotheses in a centralized setup is reviewed. Afterwards, the Consensus+Innovations Matrix Sequential Probability Ratio Test ($\mathcal{CI}$MSPRT) is proposed as a fusion of the $\mathcal{CI}$SPRT and the MSPRT. Furthermore, it is shown how the expected runlength of the algorithm can be accurately approximated. In a next step, robust versions of the $\mathcal{CI}$SPRT are developed based on LFDs and robust estimators. The performance of the proposed algorithms is verified and evaluated in the simulation section.

Chapter 6 considers the alternative paradigm of evidence-based hypothesis testing in a sequential and distributed manner. After formulating the problem, an overview of the Dempster-Shafer Theory of Evidence (DST) is given. Subsequently, a distributed sequential detector based on this theory is proposed, which uses a novel approach for performing the basic probability assignment (BPA). Furthermore, methods for robustifying the algorithm against outliers are presented. The performance of the proposed methods is validated and evaluated through simulations.

Chapter 7 draws conclusions and provides an outlook into possible future research directions.

Part II: Location Estimation & Tracking

The second part of this work is concerned with the estimation and subsequent tracking of location parameters in the example of distributed multi-target tracking.

Chapter 8 provides an introduction into the problem of multi-target tracking. First, the data association issue is discussed. Subsequently, the considered state-space and measurement models are introduced and a review of the theory behind random finite sets is given. Finally, an overview of the PHD and the PHD Filter is provided.

In Chapter 9, a novel distributed multi-tracking algorithm based on Particle PHD filtering—the Diffusion Particle PHD Filter (D-PPHDF)—is proposed and its computational complexity as well as its communication load are analyzed. Furthermore, a method for robustifying the D-PPHDF based on robust estimators is introduced and two robust variants are developed.

A centralized version of the D-PPHDF tracker—the Multi-Sensor Particle Probability Hypothesis Density Filter (MS-PPHDF)—is developed in Chapter 10. Again, an analysis of its computational complexity and communication load is performed.

Chapter 11 is dedicated to the performance evaluation of the proposed tracking algorithms. Simulations are run to compare them to an existing distributed tracker in different Gaussian and non-Gaussian environments. Furthermore, a distributed version of the Posterior Cramér-Rao Lower Bound (PCRLB) is derived that serves as a performance benchmark.

2

Fundamentals

The two overarching themes of this work are statistical robustness and distributed sensor networks. The purpose of this chapter is to give a brief overview of both of these topics and introduce important notions and concepts needed in the two major parts of the thesis.

2.1 Statistical Robustness

Statistical robustness is an area of statistics that deals with deviations from distributional assumptions. While the assumption of a Gaussian noise environment simplifies the derivation of optimal detectors and estimators, this premise is violated in many practical applications. In this case, even slight deviations from the assumptions—e.g., in the form of outlying measurements—are enough to cause a performance degradation or even the breakdown of a nominally optimal algorithm. The aim of robust signal processing is to account for distributional uncertainties by developing algorithms that exhibit a close-to-optimal performance while tolerating a certain degree of deviation from the assumptions made. Further details on statistical robustness can be found in [Zoubir et al., 2012; Zoubir et al., 2018] and the references therein.

In the sequel, first, the outlier model used throughout the thesis is introduced. Second,

three robust estimators of location are presented. These estimators are used in the design of robust detection and tracking algorithms in Part I and Part II, respectively.

2.1.1 ε-contaminated Noise

A popular way to model measurement outliers is *ε-contamination*. Here, a fraction of the measurement set is assumed to be drawn from a contamination distribution H such that

$$P = (1 - \varepsilon)P^\circ + \varepsilon H,$$

where $0 \leq \varepsilon < 0.5$ is the contamination factor and P° denotes the nominal distribution. The corresponding probability density function (PDF) reads as

$$p = (1 - \varepsilon)p^\circ + \varepsilon h.$$

In this work, the nominal distribution is considered to be Gaussian with mean μ and variance σ^2, i.e.,

$$P^\circ = \mathcal{N}(\mu, \sigma^2).$$

Furthermore, the focus is on Gaussian contamination distributions with the same mean but a κ-times higher variance, i.e.,

$$H = \mathcal{N}(\mu, \kappa\sigma^2).$$

2.1.2 The Sample Median

The *sample mean* calculates the arithmetic average of a data set $\mathcal{M} = \{x_1, x_2, \ldots\}$ according to

$$\hat{x}^{\text{mean}} = \frac{1}{|\mathcal{M}|} \sum_{i=1}^{|\mathcal{M}|} x_i.$$

Hence, a single outlier is enough to cause a significant deviation from the expected value of the uncontaminated data set [Zoubir et al., 2012]. The sample mean is, therefore, a non-robust

location estimator.

A simple alternative to the sample mean is the *sample median*, which returns the middle value of the sorted data set. Therefore, it can handle up to 50 % outliers before breaking down. With $\boldsymbol{x}$ denoting the ordered vector of observations, the sample median reads as

$$\hat{x}^{\text{median}} = \begin{cases} \boldsymbol{x}\left(\frac{|\mathcal{M}|+1}{2}\right), & |\mathcal{M}| \text{ odd}, \\ \frac{1}{2}\left(\boldsymbol{x}\left(\frac{|\mathcal{M}|}{2}\right) + \boldsymbol{x}\left(\frac{|\mathcal{M}|}{2}+1\right)\right), & |\mathcal{M}| \text{ even}. \end{cases} \tag{2.1}$$

2.1.3 The M-estimator

M-estimators are an important class of robust estimators [Huber, 1981; Zoubir et al., 2012; Zoubir et al., 2018]. Intuitively speaking, M-estimators of location provide a weighted average of the input data with weights given by

$$W(x) = \begin{cases} \frac{\psi(x)}{x}, & x \neq 0, \\ \psi'(0), & x = 0, \end{cases}$$

where $\psi(x)$ is a score function and $\psi'(x)$ its first derivative. The purpose of the score function is to downweight the influence of outliers while weighting the meaningful data close to one. Hence, an appropriate choice of $\psi(x)$ is key in designing a robust estimator with high efficiency in the nominal case. While there is a variety of viable score functions*, this work focuses on Huber's score function, which is defined as

$$\psi_{\text{Hub}}(x) = \begin{cases} x, & |x| \leq c_{\text{Hub}}, \\ c_{\text{Hub}}\text{sign}(x), & |x| > c_{\text{Hub}}, \end{cases}$$

for some positive constant c_{Hub}.

*As a matter of fact, the sample median and the sample myriad can both be formulated as M-estimators by choosing appropriate score functions.

The M-estimate of the input data is obtained by recursively calculating

$$w_i(k) = W\left(\frac{x_i - \hat{x}^{\mathrm{M}}(k)}{\hat{\sigma}(\boldsymbol{x})}\right),$$
$$\hat{x}^{\mathrm{M}}(k+1) = \frac{\sum_{i=1}^{|\mathcal{M}|} w_i(k) x_i}{\sum_{i=1}^{|\mathcal{M}|} w_i(k)},$$

until $\frac{|\hat{x}^{\mathrm{M}}(k+1) - \hat{x}^{\mathrm{M}}(k)|}{\hat{\sigma}(\boldsymbol{x})} < \epsilon$ for a small, positive constant ϵ. The algorithm is initialized by setting $\hat{x}^{\mathrm{M}}(0) = \hat{x}^{\mathrm{median}}$ and estimating the scale using the normalized median standard deviation according to

$$\hat{\sigma}_{\mathrm{mad}}(\boldsymbol{x}) = 1.483 \cdot \mathrm{median}\left(|\boldsymbol{x} - \hat{x}^{\mathrm{median}}|\right),$$

where the median is calculated as in (2.1).

2.1.4 The Sample Myriad

The sample myriad is a type of M-estimator derived from optimality criteria in the Cauchy distribution. Compared to the Gaussian distribution, the Cauchy distribution has much heavier tails. Hence, when sampling from this distribution, the probability of drawing very large values is non-negligible—i.e., outliers can occur.

The sample myriad of order m reads as

$$\hat{x}^{\mathrm{myriad}} = \arg\min_{x} \prod_{i=1}^{|\mathcal{M}|} \left[m^2 + (x_i - x)^2\right],$$

where m is a freely tunable parameter commonly set to $m = \hat{\sigma}_{\mathrm{mad}}(\boldsymbol{x})$. Further details about this type of M-estimator can be found in [Gonzalez & Arce, 1996; Gonzalez & Arce, 2002].

2.2 Distributed Sensor Networks

A network of multiple agents or nodes can be categorized into one of three architectures: centralized, decentralized, or distributed [Akyildiz et al., 2002; Karl & Willig, 2007]. In a *centralized sensor network* there is a central processing unit called *fusion center*, which typically

collects the raw data observed at the individual agents to perform the signal processing task at hand. Ultimately, the final result is broadcast back to the agents. In such a network the agents either communicate directly with the fusion center or the data is relayed via neighboring agents. Networks of this kind suffer from the fact that the fusion center poses a single point of failure.

Decentralized sensor networks (see, e.g., [Veeravalli et al., 1993; Veeravalli, 1999]) share most of the properties of centralized sensor networks. The key difference is that the agents perform some if not all processing locally, either based entirely on their own measurements or through local interactions with their neighbors. The final decision or data fusion, however, is performed at the fusion center, which broadcasts the global result back to the agents.

In *distributed sensor networks* (see, e.g., [Sahu & Kar, 2016; Leonard & Zoubir, 2018a]) the individual nodes have enough communication and processing capabilities to perform the signal processing task entirely through local interactions with their neighbors. Since no fusion center is involved, there is no single point of failure. Furthermore, these networks are inherently redundant and fault-tolerant such that the loss of a single agent can easily be compensated [Olfati-Saber et al., 2007]. Moreover, the individual sensors are typically low-cost and can be easily deployed. In order to converge to the same global result at each agent, however, consensus-like algorithms are needed. Furthermore, the communication load emerging from such a setup has to be kept at bay.

A distributed sensor network of N agents or nodes can be modeled as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V}$ denoting the set of agents and $\mathcal{E}$ being the set of edges between these agents. The graph is *simple*—i.e., devoid of self-loops—, *undirected*—i.e., if node k is connected to node l, the reverse is also true—, and *connected*—meaning that no separate subgraphs exist. The *open* neighborhood of agent k is defined as

$$\mathcal{N}_k^o = \{l \in \mathcal{V} \mid (k, l) \in \mathcal{E}\},$$

i.e., the set of all agents to which k is connected by an edge. When agent k itself is included in the set, one speaks of the *closed* neighborhood, which is given by

$$\mathcal{N}_k = \mathcal{N}_k^o \cup \{k\}. \tag{2.2}$$

A simple way to define the open neighborhood of node k is by means of the *disc model* according to

$$\mathcal{N}_k^o = \{l \in \mathcal{V} \mid \|\boldsymbol{x}_k - \boldsymbol{x}_l\| \leq r_{\text{com}}\},$$

where $\|\cdot\|$ is the Euclidean distance and $\boldsymbol{x}_k$ denotes the location vector of node k. This definition is useful in many applications where data is broadcast and errors due to the degradation of signal strength with distance are neglected. In words, the open neighborhood of node k is defined by a communication radius r_{com} around its location. The closed neighborhood is again obtained as in (2.2).

PART I

Sequential Detection

3

Statistical Hypothesis Testing

Target detection in radar, spectrum sensing in cognitive radio, and object detection in image processing are just a few examples for detection problems that can be cast as statistical hypothesis tests. This chapter gives a brief overview of fixed-sample-size hypothesis tests in Section 3.1, sequential hypothesis tests in Section 3.2, as well as the concept of least favorable densities (LFDs) in Section 3.3, and introduces common notions needed in the following chapters. Details on fixed-sample-size tests can be found, e.g., in [Van Trees, 2004; Levy, 2008; Poor, 2013]. Further information on sequential hypothesis testing in general and the Sequential Probability Ratio Test (SPRT) in particular is given in [Wald, 1947]. The concept of LFDs is explained in [Fauß & Zoubir, 2016].

3.1 Fixed-Sample-Size Hypothesis Testing

The goal of fixed-sample-size hypothesis testing is to make a statement about some prevalent phenomenon based on a random vector $\boldsymbol{Y} = [Y_1, \ldots, Y_K]$ of size K observed over the domain $\mathcal{Y}$ [Levy, 2008; Poor, 2013]. Ultimately, a reliable decision should be made for one out of two or more hypotheses on the nature of the phenomenon. In *binary* hypothesis tests, the

decision is between the null hypothesis $\mathcal{H}_0$ and the alternative hypothesis $\mathcal{H}_1$,

$$\begin{aligned} \mathcal{H}_0 &: P = P_0, \\ \mathcal{H}_1 &: P = P_1, \end{aligned} \tag{3.1}$$

or equivalently

$$\begin{aligned} \mathcal{H}_0 &: \boldsymbol{Y} \sim P_0, \\ \mathcal{H}_1 &: \boldsymbol{Y} \sim P_1, \end{aligned}$$

which make an assumption on the true and unknown distribution P associated with the investigated phenomenon. *Multiple* hypothesis tests distinguish between more than two hypotheses, namely

$$\mathcal{H}_m : P = P_m, \qquad m \in \{0, \dots, M-1\}, \tag{3.2}$$

or equivalently

$$\mathcal{H}_m : \boldsymbol{Y} \sim P_m, \qquad m \in \{0, \dots, M-1\}.$$

This work considers binary as well as multiple hypothesis tests. The latter are formulated as multiple pairwise hypothesis tests between $\mathcal{H}_m$ and $\mathcal{H}_n$, $n \in \{0, \dots, M-1\}$ (more on this in Section 5.2). The following remarks are made with respect to the binary test where $M = 2$ but also hold for multiple pairwise hypothesis tests when replacing $\mathcal{H}_0$ and $\mathcal{H}_1$ with $\mathcal{H}_m$ and $\mathcal{H}_n$, respectively.

The design of a statistical hypothesis test involves defining a *decision rule* $\delta(\boldsymbol{y})$ that maps every possible realization vector $\boldsymbol{y} = [y_1, \dots, y_K]$ to either $\mathcal{H}_0$ or $\mathcal{H}_1$. This corresponds to partitioning the observation domain $\mathcal{Y}$ into two disjoint sets as [Levy, 2008]

$$\begin{aligned} \mathcal{Y}_0 &= \{\boldsymbol{y} \mid \delta(\boldsymbol{y}) = 0\} \\ \mathcal{Y}_1 &= \{\boldsymbol{y} \mid \delta(\boldsymbol{y}) = 1\}, \end{aligned}$$

where the latter is often referred to as *critical region* for historical reasons. Irrespective of which

decision rule is used, deciding between two hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$ involves the risk of making two types of errors. Each of these errors is associated with a probability, namely

- Probability of false alarm P_{FA} (Type I):
  the probability of deciding for $\mathcal{H}_1$ although $\mathcal{H}_0$ is true,
- Probability of misdetection P_{MD} (Type II):
  the probability of deciding for $\mathcal{H}_0$ although $\mathcal{H}_1$ is true.

Intuitively, a good decision rule should keep both quantities as low as possible. The well-known Neyman-Pearson criterion defines a decision rule that minimizes P_{MD} while bounding P_{FA} to a certain level. Specifically, the optimal decision rule $\delta^*(\boldsymbol{y})$ is of the form [Neyman & Pearson, 1933]

$$\delta^*(\boldsymbol{y}) = \begin{cases} 0, & Z(\boldsymbol{y}) < \tau, \\ 0 \text{ or } 1, & Z(\boldsymbol{y}) = \tau, \\ 1, & Z(\boldsymbol{y}) > \tau, \end{cases} \tag{3.3}$$

where the test statistic $Z(\boldsymbol{y})$ is calculated as

$$Z(\boldsymbol{y}) = \frac{p_1(\boldsymbol{y})}{p_0(\boldsymbol{y})}, \tag{3.4}$$

with p_0 denoting the PDF corresponding to P_0. Furthermore, the decision threshold $\tau > 0$ has to be chosen such that the false alarm constraint is fulfilled. The test statistic in (3.4) is a likelihood ratio which is why tests of this kind are often referred to as *likelihood* or *probability ratio tests*.

If the observations $\boldsymbol{y}$ are independent and identically distributed, the joint likelihood $p_0(\boldsymbol{y})$ can be expressed by a multiplication of the single likelihoods $p_0(y_i)$ leading to

$$Z(\boldsymbol{y}) = \frac{\prod_{i=1}^{K} p_1(y_i)}{\prod_{i=1}^{K} p_0(y_i)} = \prod_{i=1}^{K} \frac{p_1(y_i)}{p_0(y_i)}.$$

In this case, a popular choice of test statistic is the log-likelihood ratio (LLR), which is calcu-

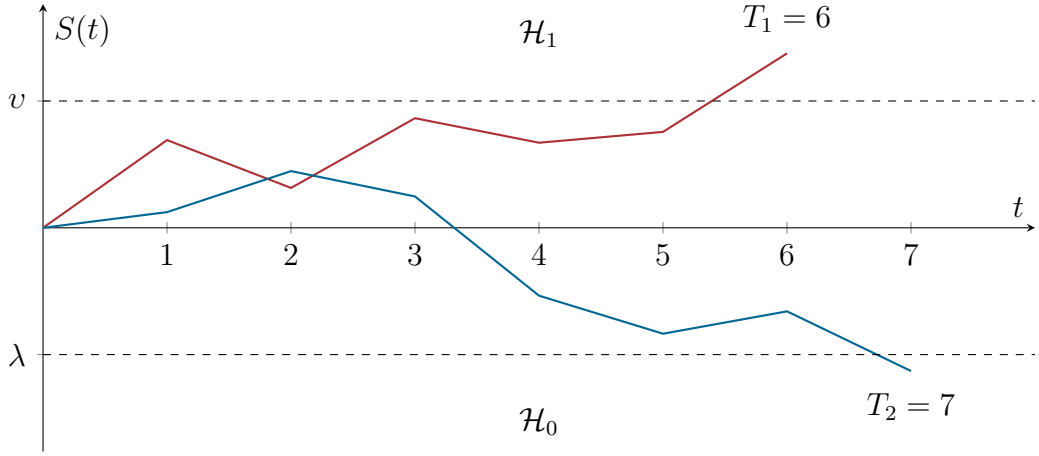

Figure 3.1: Two exemplary random walks of a test statistic in an SPRT with corresponding runlengths.

lated according to

$$S(\boldsymbol{y}) = \log(Z(\boldsymbol{y})) = \sum_{i=1}^{K} \log\left(\frac{p_1(y_i)}{p_0(y_i)}\right). \tag{3.5}$$

The decision rule (3.3) becomes

$$\delta^*(\boldsymbol{y}) = \begin{cases} 0, & S(\boldsymbol{y}) < \log(\tau)\,, \\ 0 \text{ or } 1, & S(\boldsymbol{y}) = \log(\tau)\,, \\ 1, & S(\boldsymbol{y}) > \log(\tau)\,. \end{cases}$$

3.2 Sequential Hypothesis Testing

In sequential hypothesis testing the goal is to make a reliable decision between $\mathcal{H}_0$ and $\mathcal{H}_1$ based on as few measurements as possible. The most widely used hypothesis test of this kind is the Sequential Probability Ratio Test (SPRT) as introduced by Wald [Wald, 1947]. The idea is to update the test statistic (3.5) sequentially over time as new samples are taken instead of calculating it from a fixed number of samples. The decision threshold τ is then replaced by an upper threshold υ and a lower threshold λ. These thresholds span a corridor in which the test statistic performs a random walk as shown in Figure 3.1. The decision and stopping rule

of the SPRT reads as

$$\delta(t) = \begin{cases} 0, & S(t) \leq \lambda, \\ \text{continue sampling}, & S(t) \in (\lambda, \upsilon)\,, \\ 1, & S(t) \geq \upsilon, \end{cases} \tag{3.6}$$

where

$$S(t) = \sum_{i=1}^{t} \log\left(\frac{p_1(y_i)}{p_0(y_i)}\right) = S(t-1) + \log\left(\frac{p_1(y_t)}{p_0(y_t)}\right). \tag{3.7}$$

Wald showed that performing the test in this way can reduce the required number of samples by up to 50 % on average while retaining the same error probabilities. He furthermore provided simple bounds for the decision thresholds, namely

$$\lambda \leq \frac{1-\beta}{\alpha} \qquad \text{and} \qquad \upsilon \geq \frac{\beta}{1-\alpha},$$

which solely depend on the prespecified bounds on the probabilities of false alarm and misdetection α and β, respectively.

Since in a sequential hypothesis test the number of samples is not fixed, the average runlength (ARL) is—together with P_{FA} and P_{MD}—an important metric for evaluating the performance of a sequential detector. The runlength T of the SPRT is defined as the first time instant where either $\mathcal{H}_0$ or $\mathcal{H}_1$ is accepted, i.e.,

$$T = \inf\{t \mid S(t) \notin (\lambda, \upsilon)\}\,.$$

Wald showed that—when neglecting the possible overshoot of the test statistic upon crossing a threshold—the *expected* runlength of the test can be approximated by

$$\mathbb{E}_0[T] \approx \frac{1}{D(p_1 \mid p_0)} \left(\alpha \log(\upsilon) + (1-\alpha)\log(\lambda)\right),$$
$$\mathbb{E}_1[T] \approx \frac{1}{D(p_1 \mid p_0)} \left((1-\beta)\log(\upsilon) + \beta \log(\lambda)\right),$$

where $\mathbb{E}_m[\cdot]$ denotes the expectation under $\mathcal{H}_m$. Furthermore, $D(p_1 \,|\, p_0)$ is the Kullback-Leibler divergence—a measure for the similarity of two distributions with PDFs p_0 and p_1. It is given by the expected value of the LLR, i.e.,

$$D(p_1 \,|\, p_0) = \mathbb{E}_1\left[\log\left(\frac{p_1(y)}{p_0(y)}\right)\right] = \int p_1(x) \log\left(\frac{p_1(x)}{p_0(x)}\right) \mathrm{d}x.$$

3.3 Least Favorable Densities

As discussed in Section 2.1, many practical applications suffer from an uncertainty about the distribution of the observed data. Taking this uncertainty into account transforms the *simple* hypothesis tests in (3.1) and (3.2) into *composite* tests between M disjoint sets of probability distributions $\mathcal{P}_m$ with corresponding hypotheses

$$\mathcal{H}_m : P \in \mathcal{P}_m.$$

One possible way to characterize these sets is with the help of Kassam's band model [Kassam, 1981; Fauß & Zoubir, 2016] as

$$\mathcal{P}_m = \left\{ P_m \,\middle|\, p'_m \leq p_m \leq p''_m \right\}, \tag{3.8}$$

where p'_m and p''_m are nonnegative functions whose integral over the sample space is less than one and greater than one, respectively. Note that the inequalities in (3.8) are defined pointwise. This model is flexible as it allows for varying local degrees of uncertainty on different regions of the sample space. It can either be constructed by hand based on expert knowledge, or statistically using confidence interval estimators. In contrast to many parametric uncertainty models, Kassam's model can be easily interpreted and visualized. Furthermore, it generalizes several popular uncertainty models such as the ε-contamination model as will be seen shortly. All of these reasons attest to the usefulness of this model for a wide variety of applications.

Intuitively speaking, Kassam's band model assumes the true density p_m to lie within a band specified by p'_m and p''_m, which can be interpreted as a confidence interval for p_m. A suitable decision rule in this case is one that is optimized for the worst case and, hence, guarantees a reliable performance for all possible cases. Tests of this kind are referred to as *minimax optimal*

or *minimax robust* since they minimize the maximum possible risk associated with the test (see, e.g., [Levy, 2008; Poor, 2013]). In [Fauß, 2016] a *minimax robust* version of the SPRT is introduced, which replaces the LLR in (3.7) with the LLR of the LFDs q_0 and q_1. Since these probability densities represent the worst case, a test run under these conditions will *at least* attain the performance specified by α and β. Using the algorithm provided in [Fauß & Zoubir, 2016], the LFDs for a fixed-sample-size test are iteratively calculated according to

$$\begin{aligned} q_0 &= \min\{p_0'', \max\{c_0(\nu q_0 + q_1), p_0'\}\}\,, \\ q_1 &= \min\{p_1'', \max\{c_1(q_0 + \nu q_1), p_1'\}\}\,, \end{aligned} \tag{3.9}$$

for some $\nu \geq 0$ and some $c_0, c_1 \in (0, \frac{1}{\nu}]$. Details of how to obtain LFDs that are minimax optimal in a sequential architecture can be found in [Fauß, 2016].

Uncertainties of the ε-contamination type as defined in Section 2.1.1 are included in Kassam's band model by setting $p_0' = (1-\varepsilon)p_0^\circ$, $p_0'' = p_1'' = \infty$, and $\nu = 0$, so that (3.9) reduces to

$$\begin{aligned} q_0 &= \max\{c_0 q_1, p_0'\}\,, \\ q_1 &= \max\{c_1 q_0, p_1'\}\,. \end{aligned}$$

The resulting densities correspond to the LFDs of Huber's clipped likelihood ratio test [Huber, 1965; Huber, 1981], which censors outliers and, thus, prevents them from having an unbounded effect on the test. Due to this property, it makes sense to use the centralized, fixed-sample-size LFDs also in the context of distributed sequential detection. While they are not minimax optimal in this case, they induce robustness by limiting the influence of large values at the cost of an increased ARL as will be shown in Chapter 4.

4

Robust Sequential Binary Hypothesis Testing

This chapter is concerned with binary hypothesis tests that are performed in a sequential manner in a distributed network architecture and are subject to distributional uncertainties. In Section 4.1, the problem of performing shift-in-mean and shift-in-variance tests in this setup is formulated. Section 4.2 introduces a general formulation of the Consensus+Innovations Sequential Probability Ratio Test ($\mathcal{CI}$SPRT)—a fully distributed sequential detection algorithm. The corresponding decision thresholds are derived in Section 4.3. Section 4.4 and Section 4.5 present two different paradigms for robustifying the $\mathcal{CI}$SPRT against outliers of the ε-contamination kind. The simulations in Section 4.6 verify and evaluate the performance of the proposed algorithms in a shift-in-mean and a shift-in-variance test. The findings are summarized in Section 4.7.

The contributions presented in this chapter have been published in [Leonard & Zoubir, 2017], [Hou, Leonard et al., 2017], and [Leonard & Zoubir, 2018a].

4.1 Problem Formulation

Let $\boldsymbol{Y}(t) \in \mathbb{R}^N, t = 1, 2, \ldots$ be a sequence of random vectors with entries $Y_k(t), k = 1, \ldots, N$. For all k and t the random variables $Y_k(t)$ are assumed to be independent and identically distributed according to distribution P, which admits a density p. In distributed sequential detection, each agent k sequentially performs a binary statistical hypothesis test to decide between the null hypothesis $\mathcal{H}_0$ and the alternative $\mathcal{H}_1$ according to (3.1). To this end, it takes a measurement $y_k(t)$ at time instant t from which a test statistic is computed. Considering a Gaussian environment, the hypotheses read as

$$\begin{aligned} \mathcal{H}_0 &: Y_k(t) \sim \mathcal{N}(\mu_0, \sigma_0^2), \\ \mathcal{H}_1 &: Y_k(t) \sim \mathcal{N}(\mu_1, \sigma_1^2), \end{aligned} \tag{4.1}$$

where $\mu_i, i \in \{0, 1\}$ is the known mean and σ_i^2 the variance of a zero-mean Gaussian noise process. While the results of this chapter can be applied to any binary hypothesis test of this type, the focus is on the following two test scenarios:

- Scenario 1: Shift-in-Mean Test
  The mean of the distribution under the true hypothesis is tested for, assuming equal variance σ^2 under both $\mathcal{H}_0$ and $\mathcal{H}_1$. The hypotheses become

$$\begin{aligned} \mathcal{H}_0 &: Y_k(t) \sim \mathcal{N}(\mu_0, \sigma^2), \\ \mathcal{H}_1 &: Y_k(t) \sim \mathcal{N}(\mu_1, \sigma^2). \end{aligned}$$

- Scenario 2: Shift-in-Variance Test
  The test is for the variance of the distribution under the true hypothesis assuming two zero-mean Gaussian distributions. An example for this is a test for the presence or absence of a zero-mean signal with known variance σ_x^2 in zero-mean noise with power σ_n^2, i.e.,

$$\begin{aligned} \mathcal{H}_0 &: Y_k(t) \sim \mathcal{N}(0, \sigma_n^2), \\ \mathcal{H}_1 &: Y_k(t) \sim \mathcal{N}(0, \sigma_x^2 + \sigma_n^2). \end{aligned}$$

4.2 The Consensus+Innovations Sequential Probability Ratio Test

In [Sahu & Kar, 2014; Sahu & Kar, 2016] the Consensus+Innovations Sequential Probability Ratio Test ($\mathcal{CI}$SPRT) is proposed as a distributed sequential detector based on the *consensus+innovations* approach [Kar & Moura, 2013]. In analogy to the centralized SPRT introduced by Wald [Wald, 1947], each agent k in the $\mathcal{CI}$SPRT compares its test statistic $S_k(t)$ at time instant t with an upper and a lower threshold to either decide for one of the two hypotheses if the respective threshold is crossed, or to continue the test. $S_k(t)$ is recursively calculated as

$$S_k(t) = \sum_{l \in \mathcal{N}_k} w_{kl} \left(S_l(t-1) + \eta_l(t) \right), \tag{4.2}$$

with w_{kl} denoting appropriate combination weights that sum to one. Furthermore, $\eta_l(t)$ is the LLR of node l at time instant t, which is calculated as

$$\eta_l(t) = \log\left(\frac{p_1(y_l(t))}{p_0(y_l(t))}\right) = \frac{\sigma_1^2 \left(y_l(t) - \mu_0\right)^2 - \sigma_0^2 \left(y_l(t) - \mu_1\right)^2}{2\sigma_0^2\sigma_1^2} + \log\left(\frac{\sigma_0}{\sigma_1}\right) \tag{4.3}$$

assuming the general formulation from (4.1). By collecting the combination weights into an $N \times N$ combination matrix, (4.2) can be rewritten as

$$S_k(t) = \sum_{j=1}^{t} \boldsymbol{e}_k^\top \boldsymbol{W}^{t+1-l} \boldsymbol{\eta}(j), \tag{4.4}$$

with $\boldsymbol{e}_k$ denoting the kth column of identity matrix $\boldsymbol{I}$ of size N. The LLRs of all agents at time instant j are collected in the vector $\boldsymbol{\eta}(j) = [\eta_1(j), \ldots, \eta_N(j)]^\top$. In the sequel, the choice of the weighting matrix $\boldsymbol{W}$ is discussed.

4.2.1 Choosing Weighting Matrix $\boldsymbol{W}$

In [Sahu & Kar, 2016], the authors assume a weighting matrix that is non-negative, symmetric, irreducible, and stochastic by design. However, the design process relies on a method originally introduced in [Xiao & Boyd, 2004], which can—and most of the time will—produce a

matrix with negative weights on the diagonal as explicitly stated by the authors. In the context of distributed detection, such a matrix is not practical since it will cause each node k to give its own information a negative weight. This operation has no meaning in distributed sensor networks.

Instead of requiring the weighting matrix to be doubly-stochastic, this work considers a right-stochastic matrix—i.e., its rows sum up to one. Matrices of this kind are common and well-studied, e.g., in the context of diffusion adaptation [Sayed, 2013]. An example for a right-stochastic matrix is one that puts equal weight on the information of the closed neighborhood of a node, i.e., the entries of $\boldsymbol{W}$ are given by

$$w_{kl} = \begin{cases} \frac{1}{|\mathcal{N}_k|}, & l \in \mathcal{N}_k, \\ 0, & \text{otherwise.} \end{cases}$$

Note that a right-stochastic matrix does not fulfill the requirement

$$\boldsymbol{W}^t \to \frac{1}{N}\mathbf{1}\mathbf{1}^\top,$$

where $\mathbf{1}$ is the one-vector of length N. This requirement guarantees the convergence to a networkwide consensus over time. This work, however, focuses on reaching a networkwide *decision* meaning that the individual test statistics do not have to converge to the exact same value. Averaging over the individual stopping times at each node results in a networkwide ARL of the distributed sequential detector, which is one of the performance metrics used in Section 4.6.

4.3 Decision Thresholds for the $\mathcal{CI}$SPRT

The decision thresholds derived in [Sahu & Kar, 2016] suffer from two disadvantages. First, they only hold for the specific case of symmetric Gaussian shift-in-mean hypothesis tests. In [Leonard & Zoubir, 2017] and [Hou, Leonard et al., 2017], these thresholds were generalized for use in arbitrary binary hypothesis tests. Second, the derivation of the thresholds relies on the symmetry of $\boldsymbol{W}$, an assumption that is usually not valid in distributed sensor networks. In the sequel, the generalized thresholds from [Leonard & Zoubir, 2017] and [Hou, Leonard

et al., 2017] are improved by requiring only the right-stochasticity of $\boldsymbol{W}$ in the derivation. First, however, expressions for the mean and the variance of the test statistic under $\mathcal{H}_0$ and $\mathcal{H}_1$ are derived, which are needed in the subsequent steps.

4.3.1 Mean and Variance of the Test Statistic

The expected value of the test statistic (4.2) under hypothesis $\mathcal{H}_i, i \in \{0, 1\}$ is given by

$$\begin{aligned}\mathbb{E}_i[S_k(t)] &= \sum_{j=1}^{t} \boldsymbol{e}_k^\top \boldsymbol{W}^{t+1-j} \mathbb{E}_i[\boldsymbol{\eta}(j)]\,, \\ &= \mu_{\eta,i} \sum_{j=1}^{t} \boldsymbol{e}_k^\top \boldsymbol{W}^{t+1-j} \mathbf{1}, \\ &= \mu_{\eta,i} t,\end{aligned} \tag{4.5}$$

where $\mathbb{E}_i[\cdot]$ denotes taking the expectation under hypothesis $\mathcal{H}_i$ and $\mu_{\eta,i} = \mathbb{E}_i[\boldsymbol{\eta}(j)]$ is the expected value of the LLR under $\mathcal{H}_i$.

The variance of the test statistic (4.2) under hypothesis $\mathcal{H}_i$ can be calculated as

$$\begin{aligned}\mathrm{Var}_i\left[S_k(t)\right] &= \mathbb{E}_i\left[(S_k(t) - \mu_{\eta,i} t)^2\right]\,, \\ &= \mathbb{E}_i\left[S_k(t)^2\right] - 2\mu_{\eta,i} t \mathbb{E}_i\left[S_k(t)^2\right] + \mu_{\eta,i}^2 t^2, \\ &= \mathbb{E}_i\left[S_k(t)^2\right] - \mu_{\eta,i}^2 t^2, \\ &= \mathbb{E}_i\left[\left(\sum_{j=1}^{t} \boldsymbol{e}_k^\top \boldsymbol{W}^{t+1-j} \boldsymbol{\eta}(j)\right)^2\right] - \mu_{\eta,i}^2 t^2, \\ &= \sum_{j=1}^{t}\sum_{l=1}^{t} \boldsymbol{e}_k^\top \boldsymbol{W}^{t+1-j} \mathbb{E}_i[\boldsymbol{\eta}(j)\boldsymbol{\eta}(l)] \left(\boldsymbol{W}^{t+1-l}\right)^\top \boldsymbol{e}_k - \mu_{\eta,i}^2 t^2.\end{aligned} \tag{4.6}$$

Since the expected value $\mathbb{E}_i[\boldsymbol{\eta}(j)\boldsymbol{\eta}(l)]$ can be written as

$$\mathbb{E}_i[\boldsymbol{\eta}(j)\boldsymbol{\eta}(l)] = \begin{cases} \mu_{\eta,i}^2 \mathbf{1}\mathbf{1}^\top & , j \neq l, \\ \sigma_{\eta,i}^2 \boldsymbol{I} + \mu_{\eta,i}^2 \mathbf{1}\mathbf{1}^\top & , j = l, \end{cases}$$

where $\sigma_{\eta,i}^2$ denotes the variance of the LLR under $\mathcal{H}_i$, rearranging the two sums in the last

line of (4.6) leads to

$$\begin{aligned}
\mathrm{Var}_i\left[S_k(t)\right] &= \sigma_{\eta,i}^2 \sum_{j=1}^{t} \boldsymbol{e}_k^\top \boldsymbol{W}^j \left(\boldsymbol{W}^j\right)^\top \boldsymbol{e}_k + \mu_{\eta,i}^2 \sum_{j=1}^{t} \sum_{l=1}^{t} \boldsymbol{W}^j \boldsymbol{1}\boldsymbol{1}^\top \left(\boldsymbol{W}^l\right)^\top \boldsymbol{e}_k - \mu_{\eta,i}^2 t^2, \\
&= \sigma_{\eta,i}^2 \sum_{j=1}^{t} \boldsymbol{e}_k^\top \boldsymbol{W}^j \left(\boldsymbol{W}^j\right)^\top \boldsymbol{e}_k + \mu_{\eta,i}^2 t^2 - \mu_{\eta,i}^2 t^2, \\
&= \sigma_{\eta,i}^2 \sum_{j=1}^{t} \boldsymbol{e}_k^\top \boldsymbol{W}^j \left(\boldsymbol{W}^j\right)^\top \boldsymbol{e}_k, \\
&= \sigma_{\eta,i}^2 \sum_{j=1}^{t} \boldsymbol{e}_k^\top \boldsymbol{W}^{j-m} \boldsymbol{W}^m \left(\boldsymbol{W}^\top\right)^m \left(\boldsymbol{W}^\top\right)^{j-m} \boldsymbol{e}_k,
\end{aligned} \tag{4.7}$$

with $1 \leq m \leq j$. By upper-bounding $\boldsymbol{W}^m \left(\boldsymbol{W}^\top\right)^m$ with a scaled all-ones matrix according to

$$\boldsymbol{W}^m \left(\boldsymbol{W}^\top\right)^m \leq \xi \boldsymbol{1}\boldsymbol{1}^\top,$$

where the inequality holds elementwise, (4.7) becomes

$$\mathrm{Var}_i\left[S_k(t)\right] \leq \sigma_{\eta,i}^2 \sum_{j=1}^{t} \boldsymbol{e}_k^\top \boldsymbol{W}^{j-m} \xi \boldsymbol{1}\boldsymbol{1}^\top \left(\boldsymbol{W}^\top\right)^{j-m} \boldsymbol{e}_k.$$

Using the properties

$$\begin{aligned}
\boldsymbol{W}\boldsymbol{1}\boldsymbol{1}^\top &= \boldsymbol{1}\boldsymbol{1}^\top, \\
\boldsymbol{1}\boldsymbol{1}^\top \boldsymbol{W}^\top &= \boldsymbol{1}\boldsymbol{1}^\top, \\
\boldsymbol{W}\boldsymbol{1}\boldsymbol{1}^\top \boldsymbol{W}^\top &= \boldsymbol{1}\boldsymbol{1}^\top,
\end{aligned}$$

an upper bound on the variance of the test statistic can be found as

$$\mathrm{Var}_i\left[S_k(t)\right] \leq \sigma_{\eta,i}^2 \sum_{j=1}^{t} \boldsymbol{e}_k^\top \xi \boldsymbol{1}\boldsymbol{1}^\top \boldsymbol{e}_k = \sigma_{\eta,i}^2 \xi t. \tag{4.8}$$

A suitable choice for ξ is the maximum value of the matrix $\boldsymbol{W}^m \left(\boldsymbol{W}^\top\right)^m$, i.e.,

$$\xi = \left\| \boldsymbol{W}^m \left(\boldsymbol{W}^\top\right)^m \right\|_{\max}, \tag{4.9}$$

where $\|\cdot\|_{\max}$ is the maximum norm of a matrix. Another choice for ξ is the largest eigenvalue of $\boldsymbol{W}^m \left(\boldsymbol{W}^\top\right)^m$ divided by the number of nodes N, i.e.,

$$\xi = \frac{1}{N} \lambda_{\max}\left(\boldsymbol{W}^m \left(\boldsymbol{W}^\top\right)^m\right). \tag{4.10}$$

The accuracy of this approximation can be tuned by the choice of m and thus traded off against computational load. In most distributed sensor networks, computational power is a scarce resource, which is why it makes sense to choose $m = 1$. However, if more computational power is available, a higher accuracy can be achieved by choosing a larger value for m.

The resulting expressions for the mean and the variance of $S_k(t)$ depend on the mean and the variance of the LLR. For general binary hypothesis tests according to (4.1), these quantities are given by

$$\begin{aligned} \mu_{\eta,0} &= -\frac{\mu_0^2 + \mu_1^2 - 2\mu_0\mu_1 + \sigma_0^2 - \sigma_1^2}{2\sigma_1^2} + \log\left(\frac{\sigma_0}{\sigma_1}\right), \\ \mu_{\eta,1} &= \frac{\mu_0^2 + \mu_1^2 - 2\mu_0\mu_1 + \sigma_1^2 - \sigma_0^2}{2\sigma_0^2} + \log\left(\frac{\sigma_0}{\sigma_1}\right), \end{aligned} \tag{4.11}$$

and

$$\begin{aligned} \sigma_{\eta,0}^2 &= \frac{1}{2}\left(1 + \frac{\sigma_0^4}{\sigma_1^4}\right) + (\mu_0 - \mu_1)^2 \frac{\sigma_0^2}{\sigma_1^4} - \frac{\sigma_0^2}{\sigma_1^2}, \\ \sigma_{\eta,1}^2 &= \frac{1}{2}\left(1 + \frac{\sigma_1^4}{\sigma_0^4}\right) + (\mu_0 - \mu_1)^2 \frac{\sigma_1^2}{\sigma_0^4} - \frac{\sigma_1^2}{\sigma_0^2}. \end{aligned} \tag{4.12}$$

The derivation of (4.11) and (4.12) is detailed in Appendix A.1.

4.3.2 Almost Surely Finite Stopping Time

Before the thresholds are derived, the test is shown to terminate almost surely under both hypotheses at finite stopping time or runlength T with

$$T = \inf \{t \mid S_k(t) \notin (\lambda, \upsilon)\},$$

where λ and υ denote the lower and upper decision threshold, respectively. Following the line of argument in [Sahu & Kar, 2016], the probability of stopping after time instant t can be upper-bounded under $\mathcal{H}_0$ by

$$P_0(T > t) \leq \mathcal{Q}\left(\frac{\lambda - \mu_{\eta,0} t}{\sigma_{\eta,0}\sqrt{\xi t}}\right)$$

so that

$$\lim_{t\to\infty} P_0(T > t) = 0,$$
$$P_0(T < \infty) = 1.$$

Here the function $\mathcal{Q}(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{u^2}{2}} \mathrm{d}u$ denotes the right tail-probability of the standard normal distribution. The proof under the alternative hypothesis is analogous.

4.3.3 Derivation of the Decision Thresholds

When the sequential test at node k stops at time t, a decision is made according to

$$\delta(S_k(t)) = \begin{cases} 0, & S_k(t) \leq \lambda, \\ 1, & S_k(t) \geq \nu. \end{cases}$$

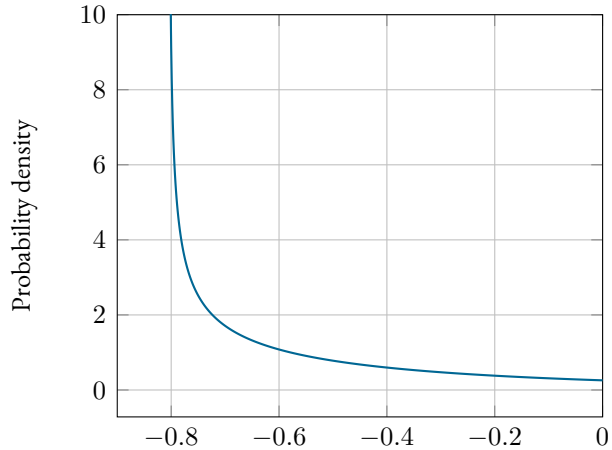

Figure 4.1: Probability density function (PDF) of the log-likelihood ratio (LLR) in a sequential binary shift-in-variance test.

Since T is finite with probability one, $S_k(T)$ is well-defined and the probability of false alarm at each node can be written as

$$\begin{aligned} P_{\text{FA}} &= P_0(S_k(T) \geq \upsilon), \\ &\leq \sum_{t=1}^{\infty} P_0(S_k(t) \geq \upsilon), \\ &\approx \sum_{t=1}^{\infty} \mathcal{Q}\left(\frac{\upsilon - \mu_{\eta,0} t}{\sigma_{\eta,0}\sqrt{\xi t}}\right). \end{aligned} \tag{4.13}$$

Inequality (4.13) holds true as long as the test statistic follows a Gaussian distribution. This is always the case in a shift-in-mean setup since the LLR is also Gaussian distributed. For shift-in-variance tests, the LLR follows a chi-squared distribution with one degree of freedom as shown in Figure 4.1. Hence, it follows from the central limit theorem that (4.13) is approximately true after just a few time steps as depicted in Figure 4.2. Further details on this are provided in Section 4.5.2.

Using the property $\mathcal{Q}(x) \leq \frac{1}{2}e^{-\frac{x^2}{2}}$ and taking a similar approach as the authors in [Sahu

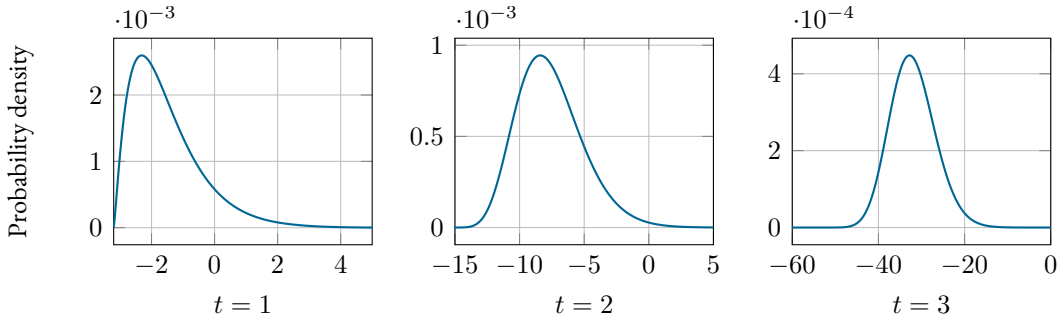

Figure 4.2: Shift-in-variance test: evolution over time of the PDF of the test statistic $S_k(t)$ of an agent with three neighbors.

& Kar, 2016], the probability of false alarm is found to be upper-bounded by

$$P_{\text{FA}} \leq \frac{2e^{\frac{\upsilon \mu_{\eta,0}}{4\sigma^2_{\eta,0}\xi}}}{1 - e^{-\frac{\mu^2_{\eta,0}}{2\sigma^2_{\eta,0}\xi}}}.$$

Requiring $P_{\text{FA}} \leq \alpha$ and solving for υ yields the upper threshold

$$\upsilon \geq \frac{4\sigma^2_{\eta,0}\xi}{\mu_{\eta,0}} \left[\log\left(\frac{\alpha}{2}\right) + \log\left(1 - e^{-\frac{\mu^2_{\eta,0}}{2\sigma^2_{\eta,0}\xi}}\right) \right]. \tag{4.14}$$

Repeating the same procedure for the probability of misdetection and requiring $P_{\text{MD}} \leq \beta$ yields the lower threshold

$$\lambda \leq \frac{4\sigma^2_{\eta,1}\xi}{\mu_{\eta,1}} \left[\log\left(\frac{\beta}{2}\right) + \log\left(1 - e^{-\frac{\mu^2_{\eta,1}}{2\sigma^2_{\eta,1}\xi}}\right) \right]. \tag{4.15}$$

The complete derivation of these expressions is deferred to Appendix A.2.

As mentioned in Remark 7.1 of [Sahu & Kar, 2016], the resulting thresholds pose only sufficient conditions on the probability of false alarm and misdetection. While tighter thresholds

can be obtained by numerically solving

$$\begin{aligned} \frac{1}{2}\sum_{t=1}^{\infty} e^{\frac{-\lambda^2-\mu_{\eta,1}^2 t^2+2\lambda\mu_{\eta,1}t}{2\sigma_{\eta,1}^2\xi t}} &= \beta, \\ \frac{1}{2}\sum_{t=1}^{\infty} e^{\frac{-\upsilon^2-\mu_{\eta,0}^2 t^2+2\upsilon\mu_{\eta,0}t}{2\sigma_{\eta,0}^2\xi t}} &= \alpha, \end{aligned} \tag{4.16}$$

the expressions in (4.14) and (4.15) are easily tractable.

4.4 A Robust Version of the $\mathcal{CI}$SPRT Based on Least Favorable Densities

In this section, the concept of LFDs is used to modify the $\mathcal{CI}$SPRT such that it can deal with composite hypotheses arising from distributional uncertainties.

4.4.1 The Robust Test Static and Its Density

In order to design a robust version of the $\mathcal{CI}$SPRT, the LLR $\eta_k(t)$ of agent k at time instant t in (4.2) is replaced by the corresponding clipped LLR

$$\eta_k^{\text{clipped}}(t) = \log\left(\frac{q_1(y_k(t))}{q_0(y_k(t))}\right), \tag{4.17}$$

i.e., the LLR of the corresponding LFDs. This yields the robust test statistic

$$\check{S}_k(t) = \sum_{l\in\mathcal{N}_k} w_{kl}\left(\check{S}_l(t-1) + \eta_l^{\text{clipped}}(t)\right). \tag{4.18}$$

Figure 4.3 compares the PDFs of the LLR and the clipped LLR side by side. By replacing the nominal densities with the LFDs, the PDF is clipped at $C_0 = -\log(c_0)$ and $C_1 = \log(c_1)$. That way, outliers are censored, which prevents them from having an unbounded influence on the test. The excess probability that accumulates at the clipping points can be calculated

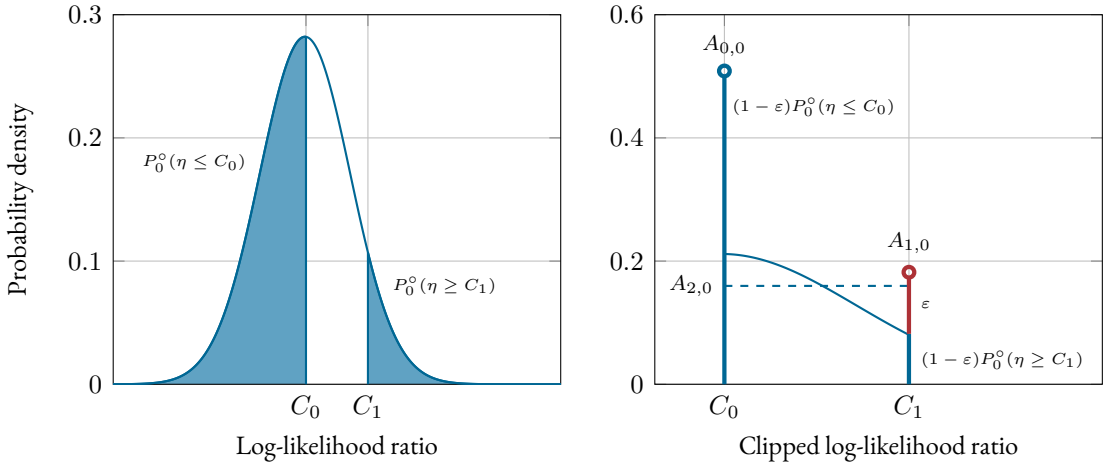

Figure 4.3: PDF of the LLR (left) and the clipped LLR (right) under $\mathcal{H}_0$. The PDF of the LLR is clipped at C_0 and C_1 and the excess probability accumulates at the clipping points. When ε-contamination is considered, the probability mass is weighted by $(1-\varepsilon)$ and the probability of drawing an outlier is added to C_1. The region between the clipping points can be approximated by a weighted uniform distribution. Under $\mathcal{H}_1$, the PDFs are mirrored at the origin.

as

$$\begin{aligned} A_{0,i} &= Q_i(\eta_k(t) \leq C_0), \\ &= (1-\varepsilon)P_i^\circ(\eta_k(t) \leq C_0) + i\varepsilon, \\ &\approx (1-\varepsilon)\mathcal{Q}\left(-\frac{C_0 - \mu_{p_i^\circ}}{\sigma^2_{p_i^\circ}}\right) + i\varepsilon, \end{aligned}$$

and

$$\begin{aligned} A_{1,i} &= Q_i(\eta_k(t) \geq C_1), \\ &= (1-\varepsilon)P_i^\circ(\eta_k(t) \geq C_1) + (1-i)\varepsilon, \\ &\approx (1-\varepsilon)\mathcal{Q}\left(\frac{C_1 - \mu_{p_i^\circ}}{\sigma^2_{p_i^\circ}}\right) + (1-i)\varepsilon, \end{aligned}$$

where $\mu_{p_i^\circ}$ and $\sigma^2_{p_i^\circ}$ denote the mean and the variance of the nominal distribution under $\mathcal{H}_i$, $i \in \{0,1\}$. When considering ε-contamination as defined in Section 2.1.1, the probability mass has to be scaled by $(1-\varepsilon)$ and the probability of drawing an outlier—denoted by ε—has to

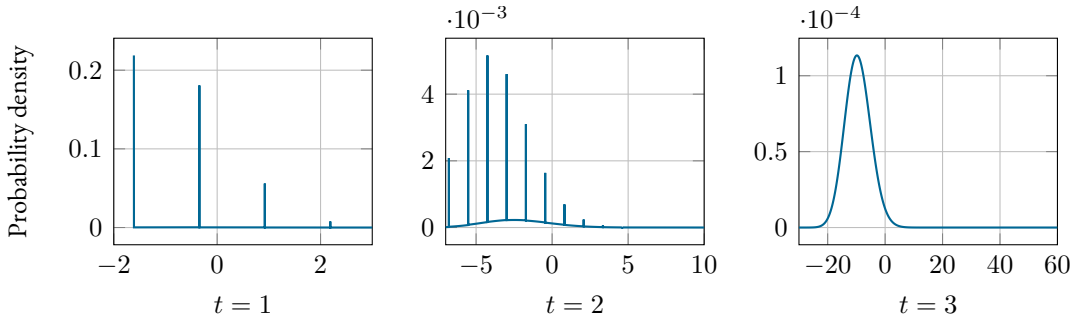

Figure 4.4: Evolution of the PDF of the robust test statistic $\check{S}_k(t)$ of an agent with three neighbors over time.

be considered. In the worst case, which is represented by the LFDs, this probability is placed at the maximum of the LLR under $\mathcal{H}_0$ and at the minimum under $\mathcal{H}_1$—i.e., at C_1 and C_0, respectively.

4.4.2 Mean and Variance of the Robust Test Statistic

The mean and the variance of the robust test statistic can be calculated by finding expressions for the mean $\mu_{\eta^{\text{clipped}},i}$ and the variance $\sigma^2_{\eta^{\text{clipped}},i}$ of the clipped LLR under $\mathcal{H}_i$, first. Since the distribution is equal for all agents, the superscript k is dropped in the following derivation. One convenient way to approximate the PDF of the clipped LLR is by two weighted Kronecker deltas at C_0 and C_1 with a weighted uniform distribution in between as shown on the right side of Figure 4.3. The mean $\mu_{\eta^{\text{clipped}},i}$ and the variance $\sigma^2_{\eta^{\text{clipped}},i}$ of the robust LLR are calculated according to

$$\begin{aligned}
\mu_{\eta^{\text{clipped}},i} &= \mathbb{E}_i\left[\eta^{\text{clipped}}\right], \\
&= \int_{\mathcal{Y}} p_{\eta^{\text{clipped}},i}(x) x \, \mathrm{d}x, \\
&= \int_{C_0}^{C_1} \left(A_{0,i}\delta(x - C_0) + A_{1,i}\delta(x - C_1) + A_{2,i}\right) x \, \mathrm{d}x, \\
&= A_{0,i}C_0 + A_{1,i}C_1 + \frac{1}{2}\left[A_{2,i}x^2\right]_{C_0}^{C_1}, \\
&= A_{0,i}C_0 + A_{1,i}C_1 + A_{2,i}\frac{C_1^2 - C_0^2}{2},
\end{aligned} \tag{4.19}$$

and

$$\begin{aligned}\sigma^2_{\eta^{\text{clipped}},i} &= \mathbb{E}_i\left[\left(\eta^{\text{clipped}}\right)^2\right] - \mu^2_{\eta^{\text{clipped}},i},\\ &= \int_{\mathcal{Y}} p_{\eta^{\text{clipped}},i}(x)x^2 \,\mathrm{d}x - \mu^2_{\eta^{\text{clipped}},i},\\ &= \int_{C_0}^{C_1} \left(A_{0,i}\delta(x-C_0) + A_{1,i}\delta(x-C_1) + A_{2,i}\right)x^2 \,\mathrm{d}x - \mu^2_{\eta^{\text{clipped}},i},\\ &= A_{0,i}C_0^2 + A_{1,i}C_1^2 + A_{2,i}\frac{C_1^3 - C_0^3}{3} - \mu^2_{\eta^{\text{clipped}},i},\end{aligned} \tag{4.20}$$

respectively, with $A_{2,i} = \frac{(1-A_{0,i}-A_{1,i})}{C_1-C_0}$.

The derivation of the mean and the variance of the non-robust test statistic $S_k(t)$ in Section 4.3.1 is based on the assumption of a Gaussian-distributed LLR $\eta_k(t)$. While this assumption does not hold for the clipped LLR $\eta_k^{\text{clipped}}(t)$, the central limit theorem can be used to show that the robust test statistic $\check{S}_k(t)$ is approximately normal for large t [Lehmann & Romano, 2005; Kay, 2006]. The evolution of the PDF of $\check{S}_k(t)$ over time is depicted in Figure 4.4 for an agent with three neighbors. As can be seen, the data exchange over the neighborhood causes the PDF to become approximately Gaussian already after the first few time instants. An even faster convergence can be observed in denser networks. Hence, the mean and variance of $\check{S}_k(t)$ can be calculated by replacing $\mu_{\eta,i}$ and $\sigma^2_{\eta,i}$ in (4.5) and (4.8) with their robust counterparts, i.e.,

$$\mathbb{E}_i\left[\check{S}_k(t)\right] = \mu_{\eta^{\text{clipped}},i}t, \tag{4.21}$$

$$\text{Var}_i\left[\check{S}_k(t)\right] \leq \sigma^2_{\eta^{\text{clipped}},i}\xi t. \tag{4.22}$$

4.4.3 Robust Decision Thresholds

The mean and the variance of the robust test statistic $\check{S}_k(t)$ in (4.21) and (4.22) have the same form as those of the non-robust test statistic $S_k(t)$ in (4.5) and (4.8). Therefore, robust decision thresholds can be derived by following the paradigm from Section 4.3.3—i.e., by replacing the mean and the variance of the LLR $\eta_k(t)$ in (4.14) and (4.15) with those of the clipped

LLR $\eta_k^{\text{clipped}}(t)$ from (4.19) and (4.20). The resulting decision thresholds are given by

$$\begin{aligned}\check{\upsilon} &\geq \frac{4\sigma^2_{\eta^{\text{clipped}},0}\xi}{\mu_{\eta^{\text{clipped}},0}}\left[\log\left(\frac{\alpha}{2}\right)+\log\left(1-e^{-\frac{\mu^2_{\eta^{\text{clipped}},0}}{2\sigma^2_{\eta^{\text{clipped}},0}}\xi}\right)\right],\\ \check{\lambda} &\leq \frac{4\sigma^2_{\eta^{\text{clipped}},1}\xi}{\mu_{\eta^{\text{clipped}},1}}\left[\log\left(\frac{\beta}{2}\right)+\log\left(1-e^{-\frac{\mu^2_{\eta^{\text{clipped}},1}}{2\sigma^2_{\eta^{\text{clipped}},1}}\xi}\right)\right].\end{aligned} \tag{4.23}$$

Again, (4.23) provides sufficient and tractable expressions for the decision thresholds. Tighter bounds are obtained by numerically evaluating

$$\begin{aligned}\frac{1}{2}\sum_{t=1}^{\infty} e^{\frac{-\lambda^2-\mu^2_{\eta^{\text{clipped}},1}t^2+2\lambda\mu_{\eta^{\text{clipped}},1}t}{2\sigma^2_{\eta^{\text{clipped}},1}\xi t}} &= \beta,\\ \frac{1}{2}\sum_{t=1}^{\infty} e^{\frac{-\upsilon^2-\mu^2_{\eta^{\text{clipped}},0}t^2+2\upsilon\mu_{\eta^{\text{clipped}},0}t}{2\sigma^2_{\eta^{\text{clipped}},0}\xi t}} &= \alpha.\end{aligned}$$

4.5 Robust Versions of the $\mathcal{CI}$SPRT Based on Robust Estimators

The algorithms developed in this section leverage the diversity of distributed sensor networks along with robust estimators to introduce robustness through the update equation of the $\mathcal{CI}$SPRT. To this end, (4.2) is reformulated as

$$S_k(t) = \sum_{l\in\mathcal{N}_k} w_{kl}\left(S_l(t-1)\right) + \hat{\eta}_k(t), \tag{4.24}$$

with $\hat{\eta}_k(t)$ denoting the weighted average of the collective innovation terms of node k and its neighborhood at time t. When no *a priori* knowledge about the reliability of the nodes is available, a common choice is to weight all the information equally. This leads to $\hat{\eta}_k(t)$ being the sample mean—a non-robust estimator of location—as defined in Section 2.1.2. Since the update equation is recursive, replacing the sample mean with a robust alternative robustifies the consensus part as well and, thus, yields a test statistic that can handle distributional un-

certainties. An advantage of introducing robustness in this manner instead of using LFDs as detailed in the previous section is the fact that the censoring takes place one stage later. Instead of clipping the LLR directly, the effect of large values on the innovation term is bounded by using a robust estimator in the combination rule. Thus, the thresholds and decision rules of the original $\mathcal{CI}$SPRT—which are based on the mean and the variance of the LLR—remain approximately valid. The interplay of fully distributed communication, consensus-like averaging of neighborhood information, and robustification of the test make a comprehensive analysis of the statistical properties of $\hat{\eta}_k(t)$ and their effects on the decision thresholds an intricate task that is beyond the scope of this work.

In the sequel, three robust versions of the $\mathcal{CI}$SPRT based on robust estimators are introduced, namely, the Median-$\mathcal{CI}$SPRT, the M-$\mathcal{CI}$SPRT, and the Myriad-$\mathcal{CI}$SPRT. Furthermore, their suitability for different binary hypothesis tests is investigated.

4.5.1 The Median-$\mathcal{CI}$SPRT, the M-$\mathcal{CI}$SPRT, and the Myriad-$\mathcal{CI}$SPRT

A straightforward way of replacing the sample mean in (4.24) with a robust alternative is to use the sample median of the neighborhood innovations, calculated as in (2.1). The resulting algorithm is dubbed Median-$\mathcal{CI}$SPRT.

The M-$\mathcal{CI}$SPRT provides a more versatile solution by replacing the sample median with an M-estimate as detailed in Section 2.1.3.

The third robust-estimator-based version of the $\mathcal{CI}$SPRT considered in this work replaces the sample mean in (4.24) with a special class of M-estimator: the sample myriad. The details of how to calculate the myriad estimate of a set of data points are given in Section 2.1.4. The resulting algorithm is referred to as Myriad-$\mathcal{CI}$SPRT.

4.5.2 The Probability Density Function of the Log-Likelihood Ratio

This work is concerned with shift-in-mean as well as shift-in-variance tests. In order to investigate the suitability of the proposed detectors in these two cases, the PDF of the estimator input—i.e., the neighborhood innovations of node k—has to be examined more closely. Since uncontaminated measurements are assumed to be Gaussian, the measurements obtained

by node k at time t can be written as

$$y_k(t) = \mu^\circ + x(t)\sigma^\circ,$$

where $x(t) \sim \mathcal{N}(0,1)$, and μ° and σ° denote the true mean and standard deviation of $y_k(t)$. The LLR of node k as defined in (4.3) becomes

Shift-in-Mean ($\sigma_0 = \sigma_1 = \sigma$):

$$\begin{aligned}
\eta_k(t) &= \frac{\sigma_1^2\left(y_k(t)-\mu_0\right)^2 - \sigma_0^2\left(y_k(t)-\mu_1\right)^2}{2\sigma_0^2\sigma_1^2} + \log\left(\frac{\sigma_0}{\sigma_1}\right), \\
&= \frac{\left(y_k(t)-\mu_0\right)^2 - \left(y_k(t)-\mu_1\right)^2}{2\sigma^2}, \\
&= \frac{\left(\sigma^\circ x(t) + (\mu^\circ-\mu_0)\right)^2 - \left(\sigma^\circ x(t) + (\mu^\circ-\mu_1)\right)^2}{2\sigma^2}, \\
&= x(t)\frac{\sigma^\circ}{\sigma^2}\left(\mu_1-\mu_0\right) + \frac{\left(\mu^\circ-\mu_0\right)^2 - \left(\mu^\circ-\mu_1\right)^2}{2\sigma^2}, \\
&= ax(t) + b,
\end{aligned}$$

and

Shift-in-Variance ($\mu_0 = \mu_1 = \mu^\circ = 0$):

$$\begin{aligned}
\eta_k(t) &= \frac{\sigma_1^2\left(y_k(t)-\mu_0\right)^2 - \sigma_0^2\left(y_k(t)-\mu_1\right)^2}{2\sigma_0^2\sigma_1^2} + \log\left(\frac{\sigma_0}{\sigma_1}\right), \\
&= \frac{\sigma_1^2 y_k(t)^2 - \sigma_0^2 y_k(t)^2}{2\sigma_0^2\sigma_1^2} + \log\left(\frac{\sigma_0}{\sigma_1}\right), \\
&= y_k(t)^2\frac{\sigma_1^2-\sigma_0^2}{2\sigma_0^2\sigma_1^2} + \log\left(\frac{\sigma_0}{\sigma_1}\right), \\
&= \left(\sigma^\circ\right)^2 x^2(t)\frac{\sigma_1^2-\sigma_0^2}{2\sigma_0^2\sigma_1^2} + \log\left(\frac{\sigma_0}{\sigma_1}\right), \\
&= cx^2(t) + d.
\end{aligned}$$

The values of a, b, c, and d are clear from the context. Thus, in the shift-in-mean test $\eta_k(t)$ follows a Gaussian distribution. In the shift-in-variance test, however, this is not the case. Since $x(t)$ follows the standard normal distribution, $x^2(t)$ is chi-squared distributed with one degree of freedom, i.e., $x^2(t) \sim \chi_1^2$. Hence, the LLR follows a scaled and shifted χ_1^2 distribution, which is skewed.

Regarding the proposed algorithms, this has the following implication: Since the median is only a robust estimator for the mean of symmetric distributions, the Median-$\mathcal{CI}$SPRT is not suitable for general shift-in-variance problems. It might give correct detection results for certain parameter choices as can be seen in the promising simulation results from [Hou, Leonard et al., 2017], but a reliable performance for arbitrary shift-in-variance tests cannot be guaranteed. Therefore, the Median-$\mathcal{CI}$SPRT will only be considered for shift-in-mean tests moving forward.

4.6 Simulations

In this section, the performance of the generalized $\mathcal{CI}$SPRT proposed in Section 4.2 and the robust detectors developed in Section 4.4 and Section 4.5 is evaluated and compared in the face of ε-contaminated noise. First, the performance is analyzed for different values of the required probabilities of false alarm and misdetection α and β, respectively. Second, the dependence on the network connectivity is investigated. Finally, an examination of the tolerable amount of outliers is performed for each algorithm.

The simulations are based on the two different test scenarios introduced in Section 4.1. In both scenarios, a randomly generated network of $N = 20$ agents with uniformly distributed x- and y-coordinates on the interval $[0, 1]$ is used and ε-contaminated noise with a ten times higher variance is considered. The network generation process is iterated until a connected network is obtained. The first scenario is a shift-in-mean problem where the objective is to decide between the means $\mu_0 = -1$ and $\mu_1 = 1$. Here, $\sigma^2 = 2$. The second scenario is a shift-in-variance test for the presence or absence of a signal with variance $\sigma_x^2 = 4$. The noise variance is $\sigma_n^2 = 1$.

The ARL and the empirical error probabilities P_{FA} and P_{MD} serve as performance metrics. The results are averaged over $N_{\text{MC}} = 10{,}000$ Monte Carlo runs and over the network by randomly selecting the value of one agent per run as in [Sahu & Kar, 2016].

4.6.1 Required Error Probabilities

In this simulation, the required false alarm and misdetection probabilities are assumed to be equal, ranging from 10^{-3} to 10^{-1}. Agents within a radius of $r_{\text{com}} = 0.6$ are considered to be neighbors. The amount of contamination is $\varepsilon = 0.1$.

Shift-in-Mean Test

The simulation results are shown in the first row of Figure 4.5. Due to the symmetry of the problem, the results are equal under both hypotheses. It can be observed that the proposed robust algorithms meet and even fall below the required error probability while the $\mathcal{CI}$SPRT fails as the requirements get more stringent. At the same time, the Median-$\mathcal{CI}$SPRT, the M-$\mathcal{CI}$SPRT, and the Myriad-$\mathcal{CI}$SPRT exhibit a slightly lower ARL compared to the non-robust $\mathcal{CI}$SPRT. Hence, robustness does not come at the cost of a longer testing time. This is in contrast to the results from [Hou, Leonard et al., 2017], where the robustification via robust estimators came with a premium in ARL. This effect can be explained by the different decision thresholds due to the different weighting matrices. The thresholds in [Hou, Leonard et al., 2017] are tighter than the ones proposed in this work but they are derived based on certain properties of the weighting matrix that are not meaningful in distributed detection setups as discussed in Section 4.2.1. Thus, it can be concluded that in a common shift-in-mean test, the introduction of robustness through robust estimators does not increase the ARL. The LFD-$\mathcal{CI}$SPRT, in contrast, needs more than twice the testing time compared to the $\mathcal{CI}$SPRT. This is in line with the results from [Leonard & Zoubir, 2017] and [Hou, Leonard et al., 2017] and due to the fact that the LFDs optimize the algorithm for the worst case, which is not attained by the considered outlier model.

As far as the required error probabilities are concerned, all robust algorithms overachieve, i.e., they do not take advantage of the tolerable number of errors but deliver a probability of false alarm and misdetection close to zero. This can be explained by the way the decision thresholds are derived in Section 4.3.3 and Section 4.4.3. As mentioned in [Sahu & Kar, 2016], the approximations required to find a closed-form solution result in thresholds that are sufficient but not optimal.

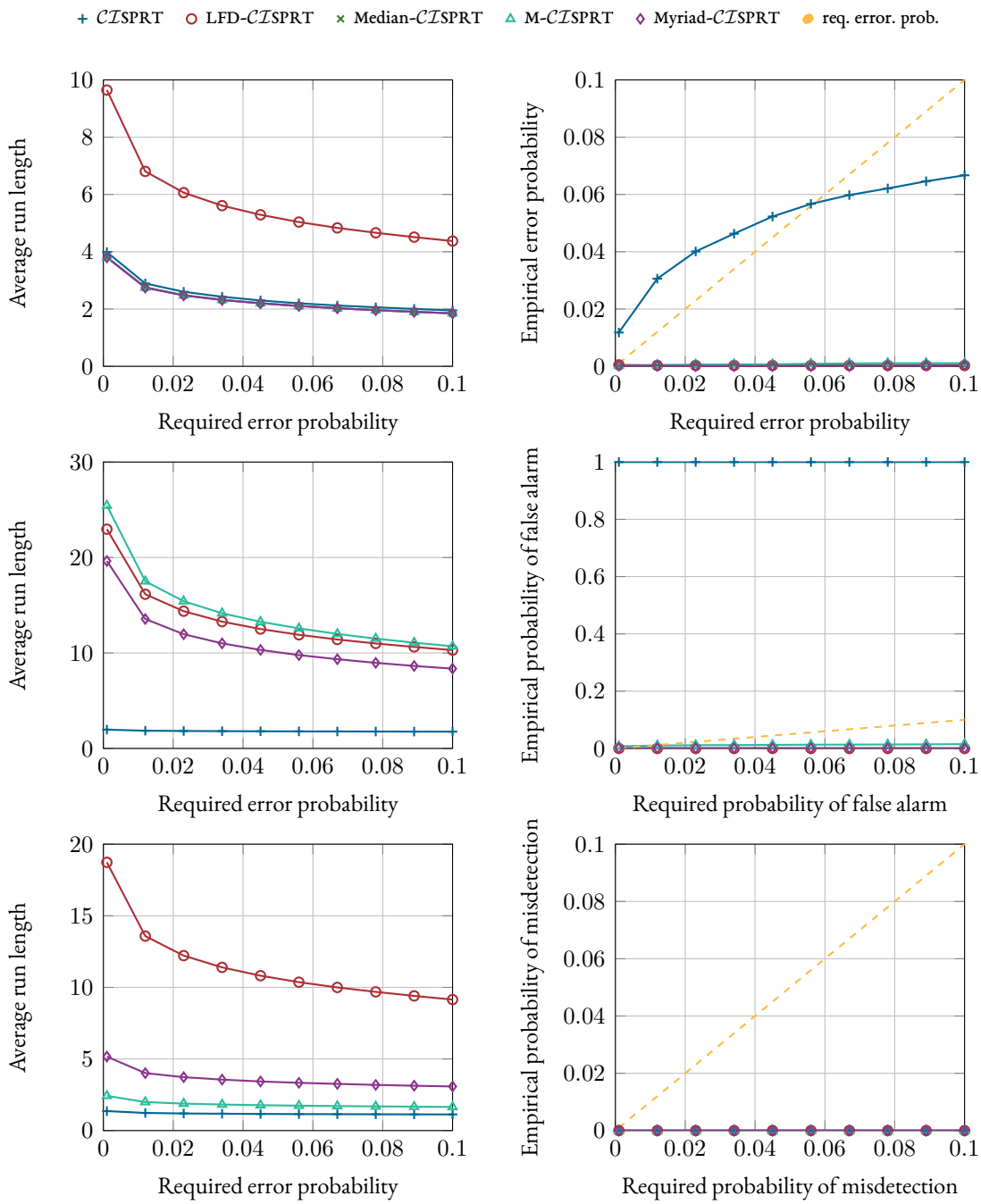

Figure 4.5: Sweep over the required probabilities of false alarm and misdetection: simulation results for the shift-in-mean test under $\mathcal{H}_0/\mathcal{H}_1$(first row) as well as the shift-in-variance test under $\mathcal{H}_0$ (second row) and $\mathcal{H}_1$ (third row). Due to the symmetry, the results for the shift-in-mean test are equal under both hypotheses.

Shift-in-Variance Test

The simulation results are shown in Figure 4.5, where the second row considers the case where $\mathcal{H}_0$ is true and the third one pertains to $\mathcal{H}_1$. Under $\mathcal{H}_0$, the $\mathcal{CI}$SPRT breaks down while all robust algorithms meet and even fall below the required error probabilities. Under $\mathcal{H}_1$, however, even the non-robust $\mathcal{CI}$SPRT meets the error requirements. This is an expected result since—in a shift-in-variance test—outliers—i.e., very large values—help in correctly deciding for $\mathcal{H}_1$. As far as the ARL is concerned, the robust algorithms exhibit a five to twelve times larger testing time in the more difficult case where $\mathcal{H}_0$ is true. Moreover, the LFD-$\mathcal{CI}$SPRT is in line with the other robust detectors, which indicates that the considered scenario approaches the worst case. Under $\mathcal{H}_1$—i.e., in the easier case—the ARL of the robust estimator approach is just a few time instants larger than that of the non-robust $\mathcal{CI}$SPRT while the LFD-$\mathcal{CI}$SPRT, again, needs considerably longer to complete the test.

4.6.2 Network Connectivity

In the following simulations, the neighborhood radius r_{com} is swept from 0.3 to 0.7 to examine the performance of the proposed algorithms for different network connectivities. The contamination ratio ε is fixed to 0.1 and the probabilities of false alarm and misdetection to 0.05.

Shift-in-Mean Test

The results under both $\mathcal{H}_0$ and $\mathcal{H}_1$ are depicted in the first row of Figure 4.6. It can be observed that the non-robust $\mathcal{CI}$SPRT exceeds the required error probability of $5\,\%$ irrespective of the neighborhood distance and is, therefore, not suitable for non-Gaussian environments. The Median-$\mathcal{CI}$SPRT, the M-$\mathcal{CI}$SPRT, and the Myriad-$\mathcal{CI}$SPRT, however, meet and even undercut the requirement while attaining the same ARL as the $\mathcal{CI}$SPRT. Hence, using robust estimators to robustify the $\mathcal{CI}$SPRT comes at no extra cost in terms of testing time. The LFD-$\mathcal{CI}$SPRT reaches an even lower error probability at the cost of an approximately three times higher testing time.

In conclusion, the communication radius—i.e., the network connectivity—does not have an influence on the robustness property of the proposed algorithms. As expected, lower net-

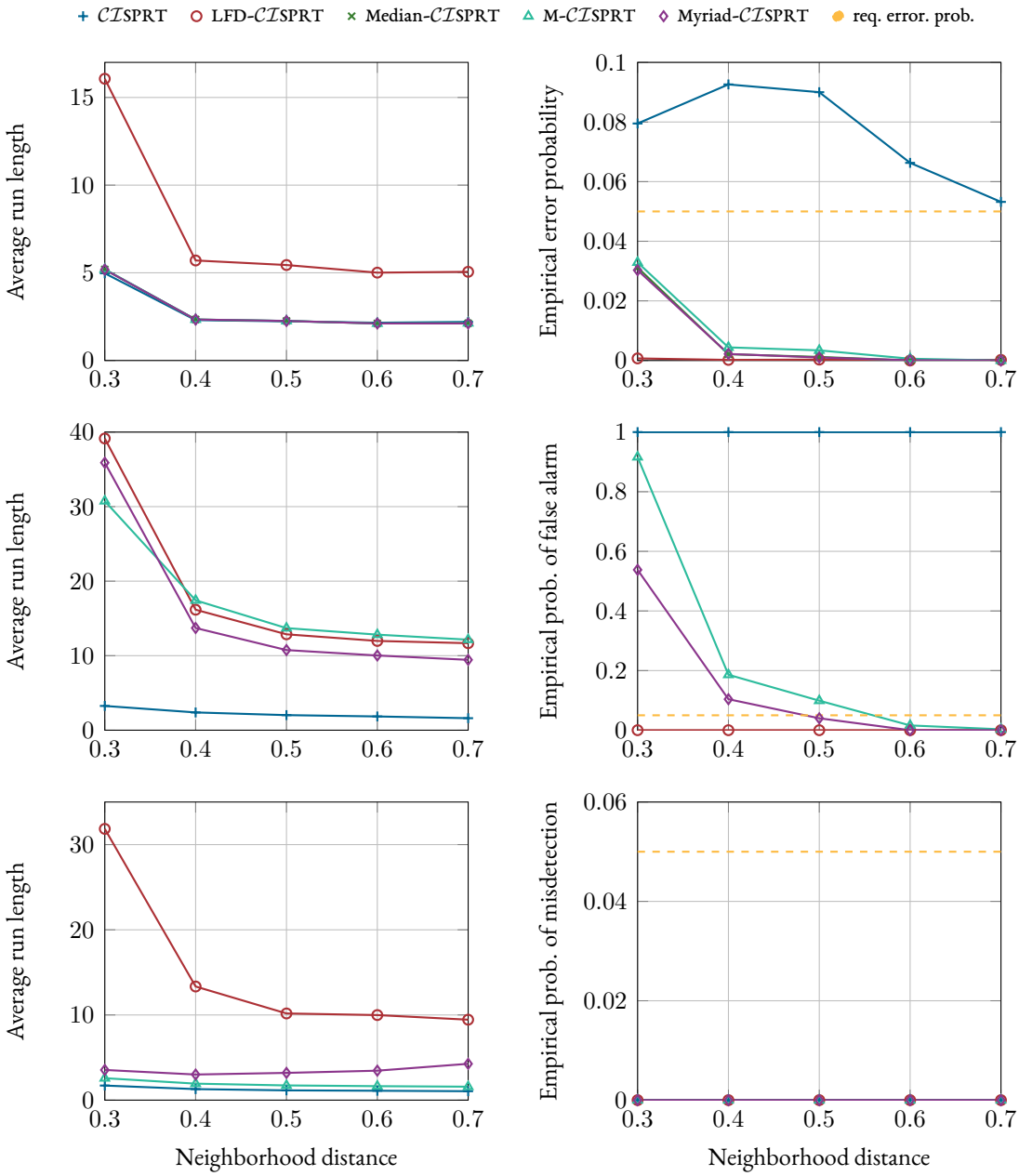

Figure 4.6: Sweep over the neighborhood radius r_{com}: simulation results for the shift-in-mean test under $\mathcal{H}_0/\mathcal{H}_1$(first row) as well as the shift-in-variance test under $\mathcal{H}_0$ (second row) and $\mathcal{H}_1$ (third row). Due to the symmetry, the results for the shift-in-mean test are equal under both hypotheses.

work connectivity leads to a higher ARL as the information dissemination over the network is slower.

Shift-in-Variance Test

The results under $\mathcal{H}_0$ and $\mathcal{H}_1$ are shown in the second and third rows of Figure 4.6, respectively. Again—since the latter is the easier test—all algorithms meet the error requirements. The robust algorithms based on robust estimators exhibit only a slight increase in the ARL while the LFD-$\mathcal{CI}$SPRT needs considerably longer to make a decision.

In the case where $\mathcal{H}_0$ is true, the non-robust $\mathcal{CI}$SPRT breaks down. The Median-$\mathcal{CI}$SPRT, the M-$\mathcal{CI}$SPRT, and the Myriad-$\mathcal{CI}$SPRT show a dependence on the network connectivity. In order to meet the required probability of false alarm, the neighborhood distance has to be larger than 0.5 or 0.6, depending on the specific algorithm. This dependence is intuitive since robust estimators need access to enough data to work reliably. The LFD-$\mathcal{CI}$SPRT, in contrast, works irrespective of the neighborhood distance, delivering a false alarm probability of 0 for all cases at approximately the same ARL as the other algorithms.

4.6.3 Amount of Contaminated Noise

In this section, the algorithms' performance is evaluated for different amounts of noise contamination. To this end, the contamination ratio ε is swept from 0 to 0.5. The neighborhood radius r_{com} is fixed to 0.6 and the probabilities of false alarm and misdetection to 0.05.

Shift-in-Mean Test

The results under $\mathcal{H}_0$ and $\mathcal{H}_1$ are given in the first row of Figure 4.7. While all algorithms work and exhibit the same ARL in the case of Gaussian noise, the $\mathcal{CI}$SPRT breaks down for $\varepsilon > 0$. The Median-$\mathcal{CI}$SPRT, the M-$\mathcal{CI}$SPRT, and the Myriad-$\mathcal{CI}$SPRT can handle up to $50\ \%$, $40\ \%$, and $30\ \%$ contamination, respectively, while meeting the required error probabilities and without an increase in testing time. The LFD-$\mathcal{CI}$SPRT works up to $\varepsilon = 0.3$ with a considerable increase in testing time. For $\varepsilon > 0.3$, the uncertainty sets overlap so that the LFDs are identical under both hypotheses. Thus, they cannot be separated.

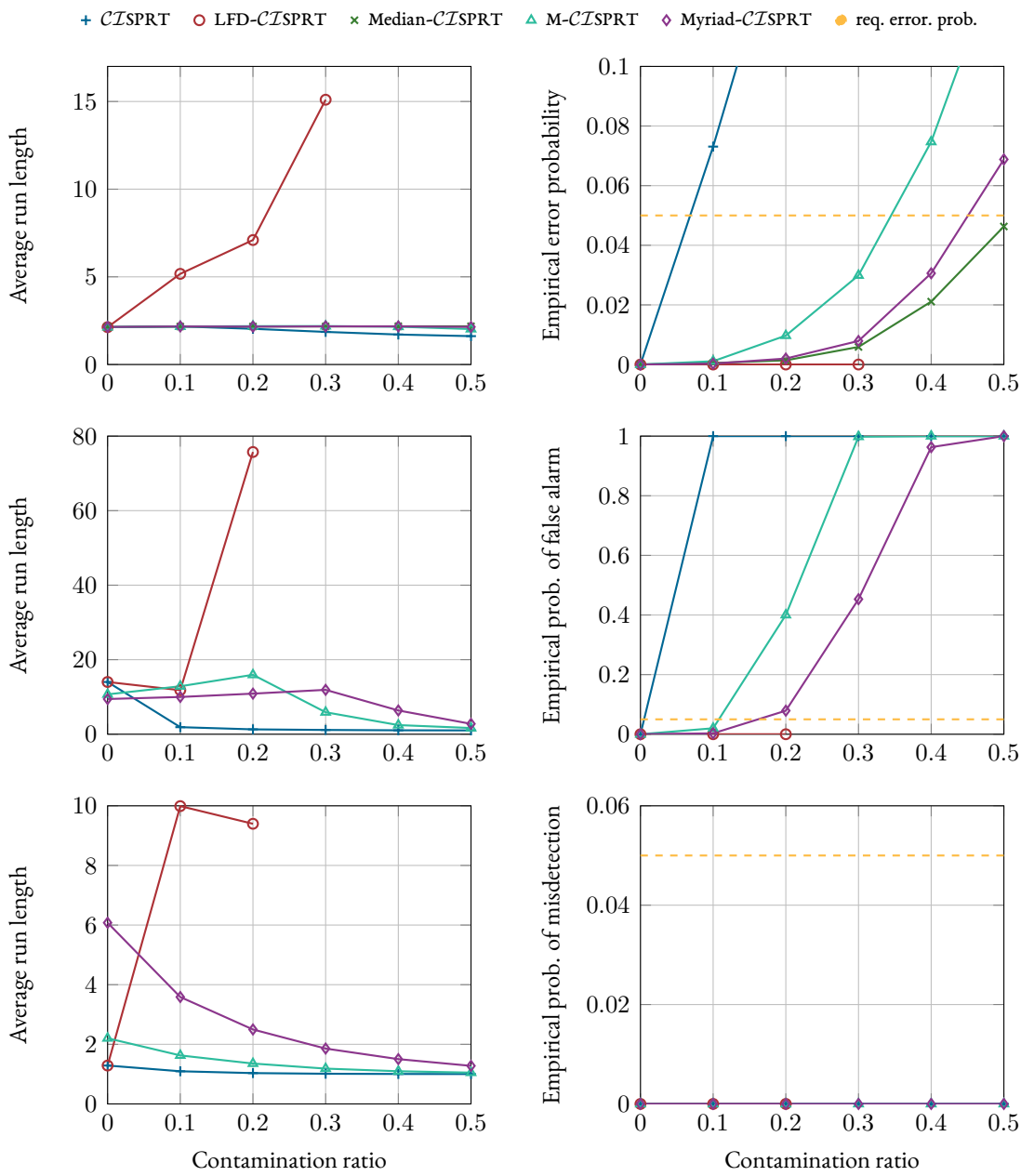

Figure 4.7: Sweep over the contamination ratio ε: simulation results for the shift-in-mean test under $\mathcal{H}_0/\mathcal{H}_1$(first row) as well as the shift-in-variance test under $\mathcal{H}_0$ (second row) and $\mathcal{H}_1$ (third row). Due to the symmetry, the results for the shift-in-mean test are equal under both hypotheses.

Shift-in-Variance Test

The results under $\mathcal{H}_0$ and $\mathcal{H}_1$ are depicted in the second and third rows of Figure 4.7, respectively. Under $\mathcal{H}_0$, the $\mathcal{CI}$SPRT breaks down for $\varepsilon > 0$. The algorithms based on robust estimators can handle 10 % contamination without an increase in testing time. The LFD-$\mathcal{CI}$SPRT is able to cope with 20 % contamination at the cost of a drastically increased ARL. For higher amounts of contamination, the hypotheses become inseparable.

Under $\mathcal{H}_1$, the LFD-$\mathcal{CI}$SPRT can only handle 20 % contamination due to the inseparability of the uncertainty sets for $\varepsilon > 0.2$. The jump in the ARL of the LFD-$\mathcal{CI}$SPRT is due to the fact that the algorithm defaults to the $\mathcal{CI}$SPRT in the case of purely Gaussian noise. Generally, the ARL decreases as the contamination increases. This is expected since a larger number of outliers helps in making the correct decision in this shift-in-variance setup.

4.7 Summary

In this chapter, a general formulation of the $\mathcal{CI}$SPRT was presented that is not only suitable for sequential binary hypothesis tests but also considers a network structure that is meaningful in the context of distributed detection. Furthermore, two paradigms for robustifying the $\mathcal{CI}$SPRT were developed and four robust sequential detection algorithms were proposed, namely, the LFD-$\mathcal{CI}$SPRT, the Median-$\mathcal{CI}$SPRT, the M-$\mathcal{CI}$SPRT, and the Myriad-$\mathcal{CI}$SPRT. After investigating their suitability for different kinds of hypothesis tests, their performance was verified, evaluated, and compared in a shift-in-mean and a shift-in-variance test. The simulation results showed that the proposed detectors are robust against outliers of the ε-contamination type at no or minimal extra cost in terms of the ARL. Only the LFD-$\mathcal{CI}$SPRT comes with a notable increase in testing time due to its focus on the worst case. Furthermore, it was shown that the robustness of the robust-estimator-based algorithms depends on the network connectivity since an estimator only works reliably if it has access to enough data. If the network is dense enough, $20 - 40$ % contamination can easily be tolerated. While the LFD-$\mathcal{CI}$SPRT can sometimes tolerate a larger number of outliers or achieve lower error probabilities, the uncertainty sets need to be separable for the LFDs to be meaningful.

5

Robust Sequential Multiple Hypothesis Testing

This chapter is concerned with robust sequential detectors for testing multiple hypotheses in a distributed sensor network. The detectors should be able to handle distributional uncertainties such as outliers to be suitable for real-life applications where the assumption of Gaussianity is violated. Section 5.1 formulates the problem of performing a multiple hypothesis shift-in-mean and shift-in-variance test in this setup. In Section 5.2, a review of the Matrix Sequential Probability Ratio Test (MSPRT) [Tartakovsky et al., 2014] for sequential multiple hypothesis testing at a centralized detector is given. In Section 5.3, the Consensus+Innovations Matrix Sequential Probability Ratio Test ($\mathcal{CI}$MSPRT) is proposed as a fusion of the MSPRT and the $\mathcal{CI}$SPRT as introduced in Section 4.2. Section 5.4 shows how to approximate the expected runlength of the algorithm. Subsequently, the $\mathcal{CI}$MSPRT is robustified in Section 5.5 using the paradigms presented in Section 4.4 and Section 4.5. The simulations in Section 5.6 verify, evaluate, and compare the performance of the proposed algorithms in shift-in-mean and shift-in-variance tests.

The contributions presented in this chapter have been published in part in [Leonard et al.,

2018b] and [Leonard et al., 2018c].

5.1 Problem Formulation

When testing multiple simple hypotheses in a distributed setup, each agent decides between $M > 1$ hypotheses of the form (3.2). As in Chapter 4, a shift-in-mean as well as a shift-in-variance scenario are considered:

- Scenario 1: Shift-in-Mean Test
  The distributions P_m have different means μ_m. Assuming zero-mean Gaussian measurement noise with variance σ^2, the hypotheses become

$$\mathcal{H}_m : Y_k(t) \sim \mathcal{N}(\mu_m, \sigma^2), \quad m = 0, \dots, M-1.$$

- Scenario 2: Shift-in-Variance Test
  P_m differ in variance σ_m^2. Hence, node k tests between

$$\mathcal{H}_m : Y_k(t) \sim \mathcal{N}(\mu, \sigma_m^2 + \sigma^2), \quad m = 0, \dots, M-1.$$

To test the robustness of the proposed algorithms, measurement outliers of the ε-contamination kind are considered as defined in Section 2.1.1.

5.2 The Matrix Sequential Probability Ratio Test

The Matrix Sequential Probability Ratio Test (MSPRT) extends the single-sensor SPRT to M hypotheses [Tartakovsky et al., 2014]. This is done by forming all possible hypothesis pairs $\mathcal{H}_m, \mathcal{H}_n$ and computing a pairwise test statistic

$$S_{mn}(t) = \sum_{i=1}^{t} \log\left(\frac{p_m(y(i))}{p_n(y(i))}\right).$$

Next, the test statistics are collected in a matrix $\boldsymbol{S}$, all entries of which are compared to a threshold matrix $\boldsymbol{\Upsilon}$ with entries

$$\upsilon_{mn} = \log\left(\frac{1-\beta_{mn}}{\alpha_{mn}}\right) \approx \log\left(\frac{1}{\alpha_{mn}}\right),$$

where α_{mn} and β_{mn} denote the respective required probabilities of false alarm and misdetection of the pairwise hypothesis test. The test is performed as an *acceptance test*, i.e., it is stopped and a decision is made in favor of $\mathcal{H}_m$ once all entries in the mth row of matrix $\boldsymbol{S}$—excluding the (m, m)th entry—surpass their corresponding thresholds in $\boldsymbol{\Upsilon}$. In mathematical terms this reads as

$$\begin{aligned} &\text{if } \exists\, m \in \{0, \dots, M-1\} \mid S_{mn}(t) \geq \upsilon_{mn} \;\forall\, n \in \{0, \dots, M-1\} \setminus \{m\} : \text{ accept } \mathcal{H}_m \\ &\text{else : continue sampling} \end{aligned} \tag{5.1}$$

where the upper threshold υ_{mn} is calculated as in (4.14). Hence, the multiple hypothesis test corresponds to performing $M(M-1)$ pairwise threshold comparisons at each time step. Note that it is also possible to perform a rejection test which rejects the $m < M$ least likely hypotheses.

5.3 The Consensus+Innovations Matrix SPRT

In order to perform multiple hypothesis testing in a distributed sensor network, the concepts of the $\mathcal{CI}$SPRT and the MSPRT are fused. In the proposed Consensus+Innovations Matrix Sequential Probability Ratio Test ($\mathcal{CI}$MSPRT) each node k first computes the LLRs for all hypothesis pairs $\mathcal{H}_m, \mathcal{H}_n$ as

$$\eta_{mn}^k(t) = \log\left(\frac{p_m(y_k(t))}{p_n(y_k(t))}\right). \tag{5.2}$$

Next, the LLRs are distributed over the neighborhood and the pairwise test statistics $S_{mn}^k(t)$ are calculated for all hypothesis pairs according to

$$S_{mn}^k(t) = \sum_{l \in \mathcal{N}_k} w_{kl} S_{mn}^l(t-1) + \sum_{l \in \mathcal{N}_k} w_{kl} \eta_{mn}^l(t). \tag{5.3}$$

Now each node k performs an acceptance test according to rule (5.1).

5.4 Expected Runlength of the $\mathcal{CI}$MSPRT

The runlength T of the $\mathcal{CI}$MSPRT is equal to the first time instant t where a hypothesis is accepted, i.e.,

$$T = \inf \left\{ t \mid S_{mn}^{k}(t) \geq \upsilon_{mn}^{u} \text{ for some } m \text{ and } \forall\, n \neq m \right\}.$$

Following the line of argument from Section 3.1, the expected runlegth of a pairwise test can be expressed with the help of the Kullback-Leibler divergence $D(p_m \mid p_n)$ as

$$\mathbb{E}_{mn}\left[T\right] \approx \frac{1}{D(p_m \mid p_n)} \left((1 - \beta_{mn}) \log\left(\upsilon_{mn}\right) + \beta_{mn} \log\left(\lambda_{mn}\right) \right).$$

Since the $\mathcal{CI}$MSPRT decides for $\mathcal{H}_m$ once the corresponding $M-1$ pairwise tests have been successful, the runlength of the entire test is dictated by the slowest of all $M-1$ pairwise tests. Hence, we define the expected runlength of the $\mathcal{CI}$MSPRT as the expected runlength of the slowest pairwise test, i.e.,

$$\mathbb{E}_{m}\left[T\right] = \max\left[\mathbb{E}_{mn}\left[T\right]\right] \;\forall\, n \in \{0, \ldots, M-1\} \text{ and } n \neq m.$$

5.5 Robust Versions of the $\mathcal{CI}$MSPRT

In order to robustify the $\mathcal{CI}$MSPRT against distributional uncertainties such as outliers, the two paradigms introduced for sequential binary hypothesis tests in Chapter 4 are applied.

The first robust version of the $\mathcal{CI}$MSPRT considered in this work is the LFD-$\mathcal{CI}$MSPRT. It is based on the concept of LFDs as detailed in Section 3.3. The test is robustified by replacing the nominal PDFs in (5.2) with the corresponding LFDs as

$$\eta_{mn}^{k,\text{clipped}}(t) = \log\left(\frac{q_m(y_k(t))}{q_n(y_k(t))}\right). \tag{5.4}$$

This operation clips the LLR at certain levels to bound the influence of outliers. Replacing the pairwise LLR in (5.3) by its clipped counterpart and performing the matrix test at each node as in (5.1), results in a robust algorithm.

Considering the paradigm of using robust estimators in the test statistic update as introduced in Section 4.5 and choosing the same estimators yields three robust versions of the $\mathcal{CI}$MSPRT, namely, the Median-$\mathcal{CI}$MSPRT, the M-$\mathcal{CI}$MSPRT, and the Myriad-$\mathcal{CI}$MSPRT. The algorithms are obtained by reformulating (5.3) as

$$S_{mn}^{k}(t) = \sum_{l \in \mathcal{N}_k} w_{kl} S_{mn}^{l}(t-1) + \hat{\eta}_{mn}^{k}(t), \tag{5.5}$$

where $\hat{\eta}_{mn}^{k}(t)$ denotes the weighted average of the collective innovations of node k and its neighborhood for the hypothesis pair $\mathcal{H}_m, \mathcal{H}_n$ at time instant t. Due to the recursive nature of the update equation, replacing the weighted average $\hat{\eta}_{mn}^{k}(t)$—which becomes the non-robust sample mean under equal weighting—automatically robustifies the entire test statistic update.

5.6 Simulations

In this section, the performance of the proposed algorithms for sequentially testing multiple hypotheses is analyzed under different network sizes $N \in \{15, 30\}$ and connectivity $r_{\text{com}} \in \{0.3, 0.6\}$. Four exemplary networks of this kind are depicted in Figure 5.1. The detectors are evaluated in the shift-in-mean and shift-in-variance scenarios defined in Section 5.1. In the shift-in-mean scenario, the test is between four hypotheses $\mathcal{H}_m, m \in \{0, \ldots, 3\}$ of different mean $\mu_m \in \{-2, -1, 1, 2\}$ and equal variance $\sigma^2 = 4$. The shift-in-variance tests are performed between four zero-mean hypotheses with $\sigma_m^2 + \sigma^2 \in \{1, 2, 4, 16\}$. Furthermore, the required probability of false alarm is fixed to $\alpha_{mn} = \alpha = 0.01$. For each hypothesis 1,000 Monte Carlo runs are performed. While it would be preferable to consider a larger number of Monte Carlo runs given the considered error probabilities, the number was chosen so as to ensure the computational complexity to be tractable. The ratio of correct detection and the ARL serve as performance metrics.

5.6.1 Results for the $\mathcal{CI}$MSPRT

First, the non-robust $\mathcal{CI}$MSPRT is evaluated in a Gaussian environment. The ARL is additionally compared to the theoretical expected runlength as calculated in Section 5.4.

The simulation results are depicted in Figure 5.2 where the upper row belongs to the shift-

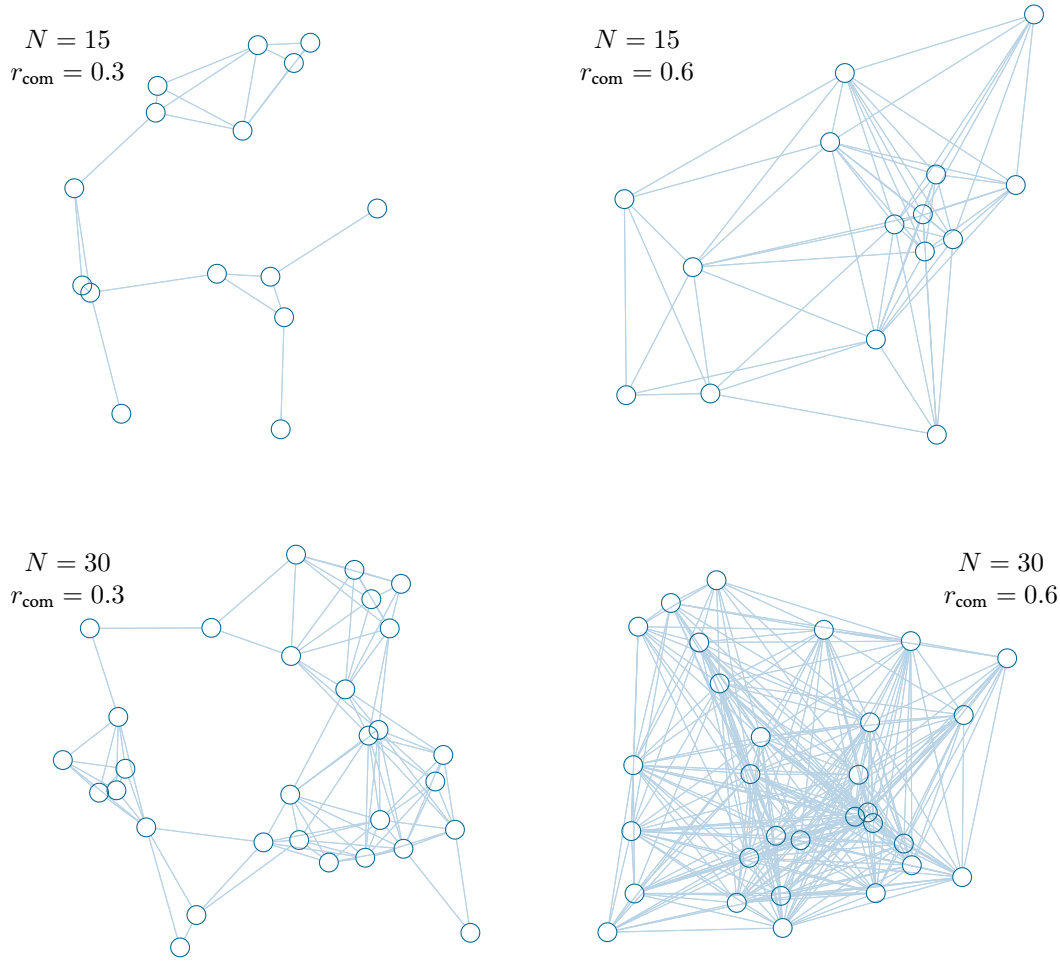

Figure 5.1: Four exemplary networks of different size $N \in \{15, 30\}$ and connectivity $r_{\text{com}} \in \{0.3, 0.6\}$.

in-mean test and the lower row shows the results for the shift-in-variance test. The $\mathcal{CI}$MSPRT clearly meets the required false-alarm probability of $\alpha = 0.01$ by delivering perfect detection results for every network configuration in both test scenarios.

In all cases, the difference between the expected runlength and the ARL is one to two time instants at maximum, which means that the performance of the algorithm can be accurately predicted beforehand. It can be observed that a higher network connectivity can drastically reduce the ARL while increasing the network size has only a marginal effect. This holds for both test scenarios and indicates a favorable scaling probability of the proposed algorithm.

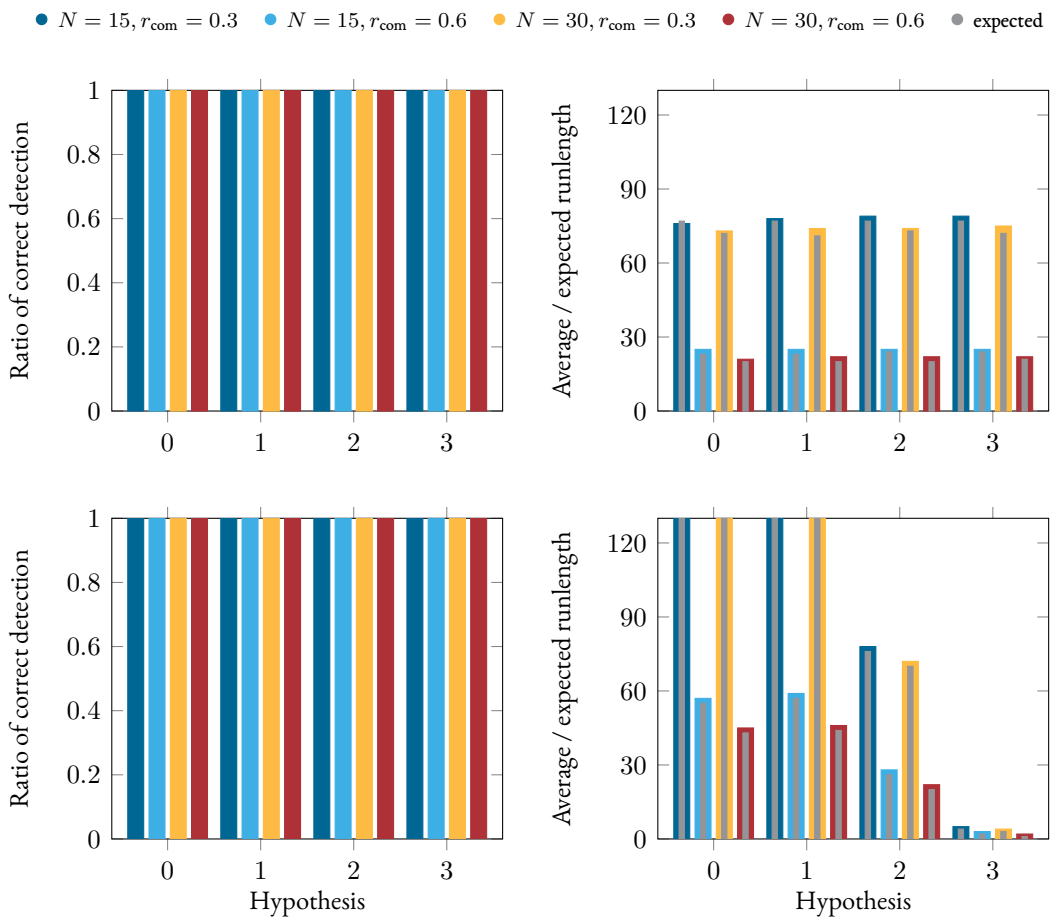

Figure 5.2: Simulation results for the $\mathcal{CI}$MSPRT in a shift-in-mean test (upper row) and a shift-in-variance test (lower row).

Furthermore, comparing the ARL of the two tests reveals that in the shift-in-mean test the decision between any two hypotheses is equally hard as visible from the equal ARL levels. This is expected and due to the symmetry of the test. The shift-in-variance test is not symmetric as confirmed by the different ARL levels. The ARL is equal under $\mathcal{H}_0$ and $\mathcal{H}_1$, and drops as the signal variance increases under $\mathcal{H}_2$ and $\mathcal{H}_3$. This is due to the fact that in a zero-mean shift-in-variance test, the overlap of the corresponding PDFs is greatest at the origin and decreases for larger measurement values. Hence, it is easier to match large measurements to the correct hypotheses.

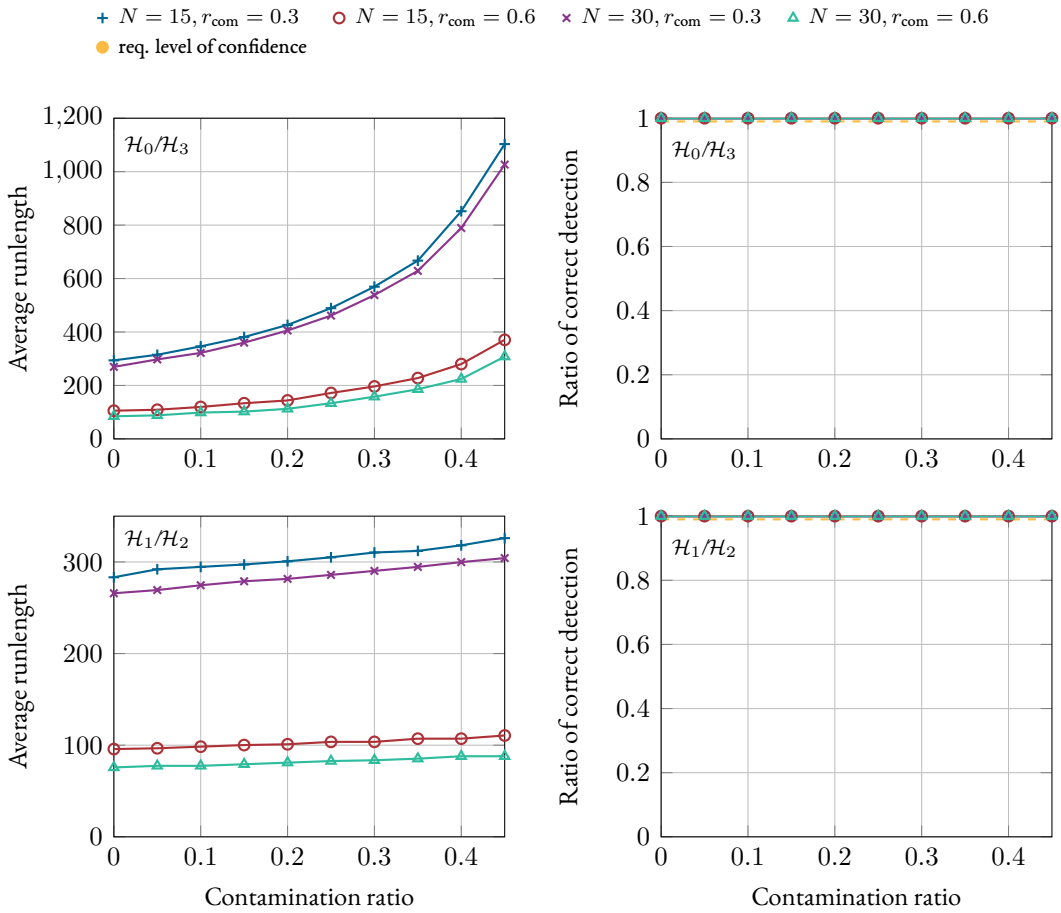

Figure 5.3: Simulation results for the LFD-$\mathcal{CI}$MSPRT in a shift-in-mean test. Due to the symmetry, equal results are obtained for $\mathcal{H}_0/\mathcal{H}_3$ and $\mathcal{H}_1/\mathcal{H}_2$.

5.6.2 Results for the Robust Versions of the $\mathcal{CI}$MSPRT

This section evaluates the performance of the robust versions of the $\mathcal{CI}$MSPRT proposed in Section 5.5. To this end, outliers of the ε-contamination type are considered and one type of contamination is assumed for all hypotheses, i.e., $h_m = h \sim \mathcal{N}(0, 81)$. The contamination ratio ε is swept over the interval $[0, 0.45]$ in the shift-in-mean test and over $[0, 0.3]$ in the shift-in-variance test.

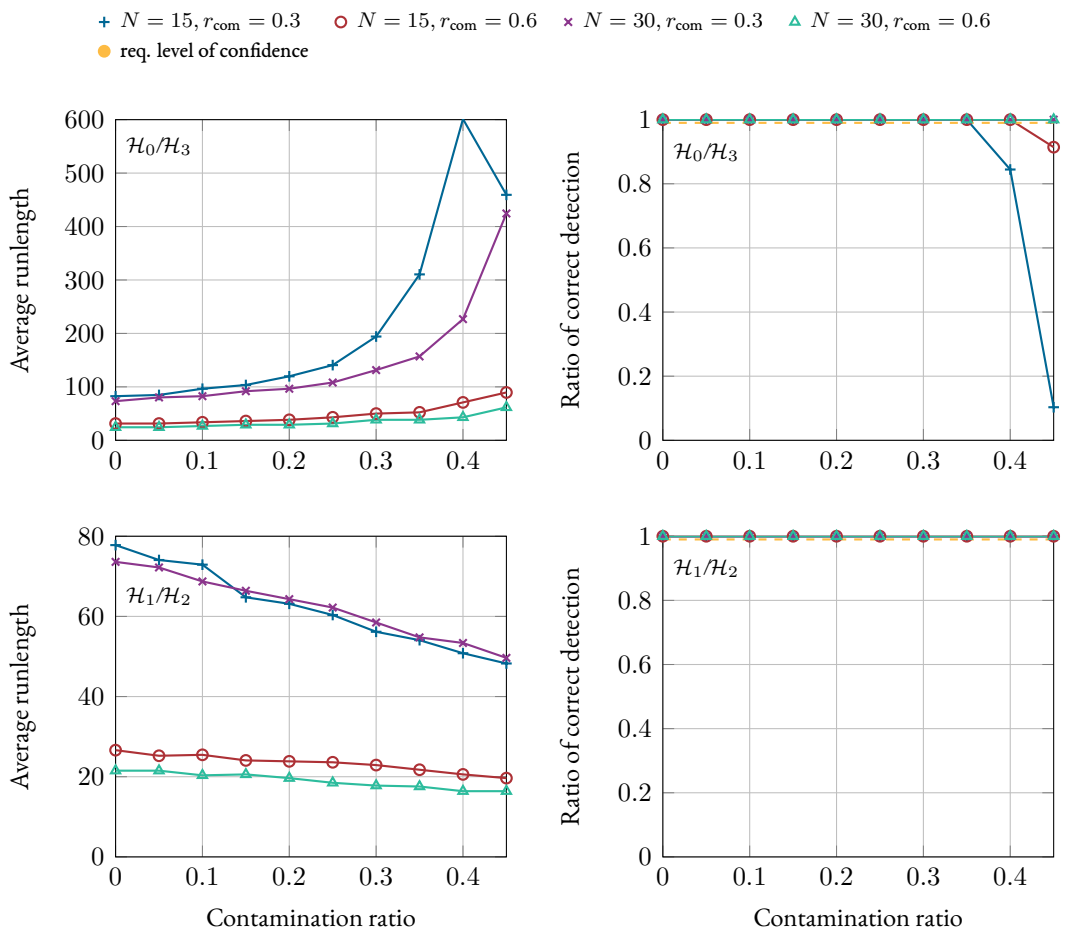

Figure 5.4: Simulation results for the Median-$\mathcal{CI}$MSPRT in a shift-in-mean test. Due to the symmetry, equal results are obtained for $\mathcal{H}_0/\mathcal{H}_3$ and $\mathcal{H}_1/\mathcal{H}_2$.

Shift-in-Mean Test

In the shift-in-mean test, the LFD-$\mathcal{CI}$MSPRT (Figure 5.3) and the three robust-estimator-based algorithms Median-$\mathcal{CI}$MSPRT (Figure 5.4), M-$\mathcal{CI}$MSPRT(Figure 5.5), and Myriad-$\mathcal{CI}$MSPRT (Figure 5.6) are compared. In the case of the M-$\mathcal{CI}$MSPRT, Huber's score function is chosen with $c_{\text{Hub}} = 1.8$. Since the test is symmetric, equal results are obtained under $\mathcal{H}_0$ and $\mathcal{H}_3$ as well as under $\mathcal{H}_1$ and $\mathcal{H}_2$ for all algorithms.

While the LFD-$\mathcal{CI}$MSPRT delivers perfect detection results under all hypotheses independently of the contamination ratio, this comes at the cost of a much higher ARL. The performance of the robust-estimator-based detectors is very similar irrespective of the chosen esti-

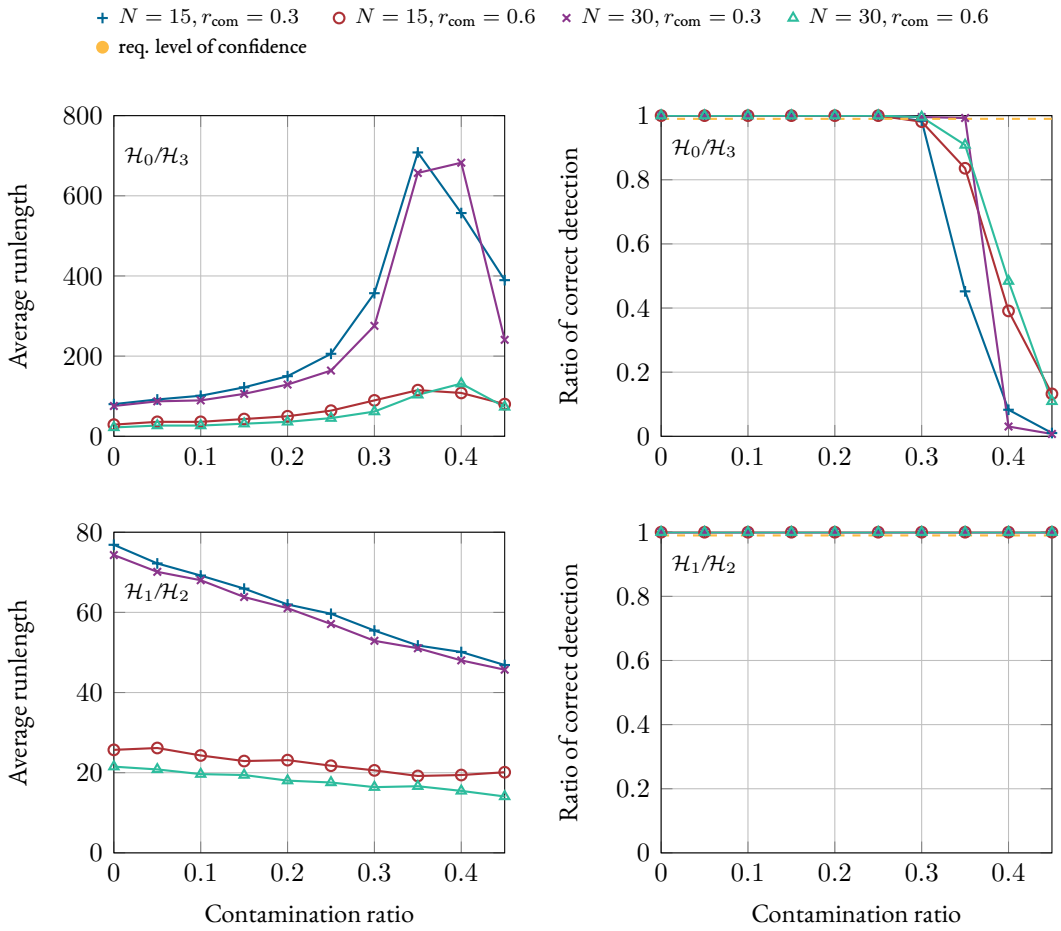

Figure 5.5: Simulation results for the M-$\mathcal{CI}$MSPRT in a shift-in-mean test. Due to the symmetry, equal results are obtained for $\mathcal{H}_0/\mathcal{H}_3$ and $\mathcal{H}_1/\mathcal{H}_2$.

mator. All of these algorithms obtain a constant ratio of correct detection of 1 only for $\mathcal{H}_1$ and $\mathcal{H}_2$. In this case, the ARL actually drops with increasing contamination since—for the two middle hypotheses with a mean close to zero—outliers help in making a correct decision. Under $\mathcal{H}_0$ and $\mathcal{H}_3$, between 25 % and 40 % contamination can be tolerated with only small differences for different estimators and network conditions. Here, the ARL increases with contamination and spikes just before the performance drop. Network size and connectivity have the same influence on all proposed algorithms, with the connectivity exhibiting a large, and the network size a small impact on the ARL.

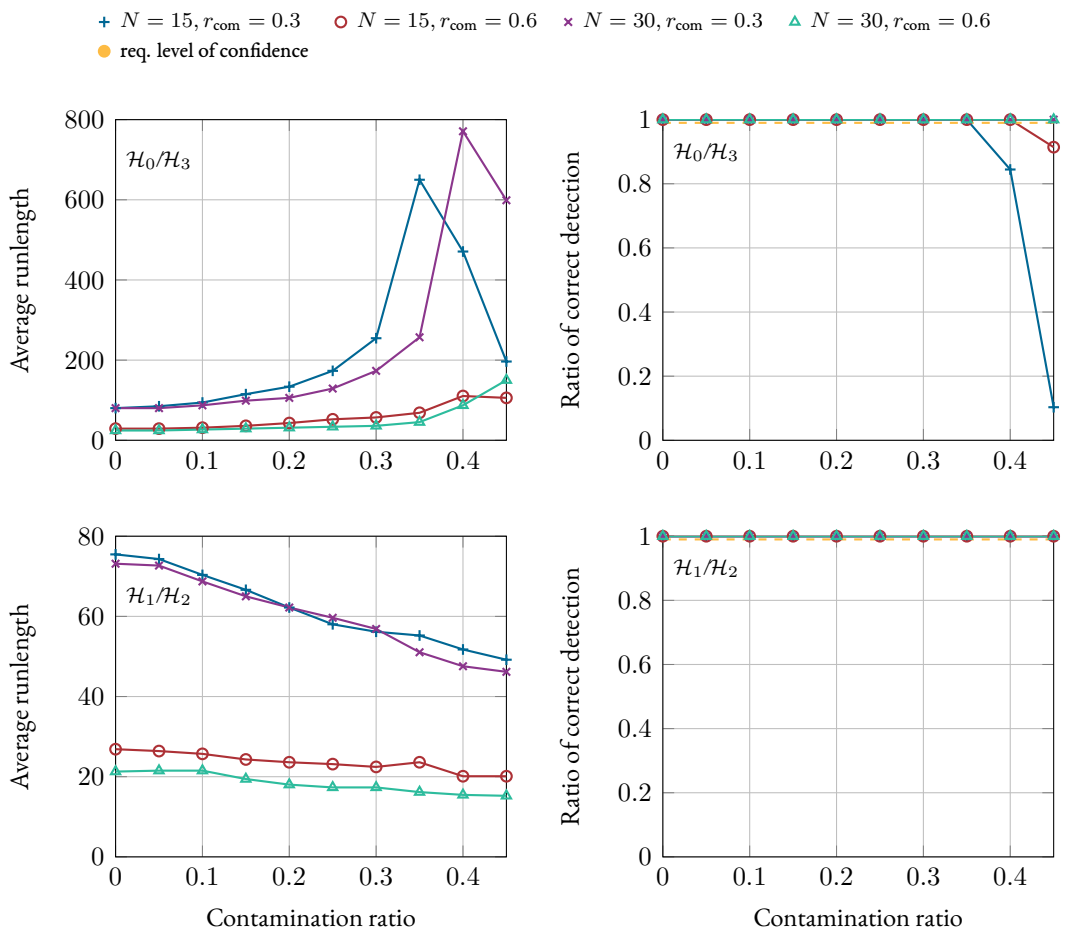

Figure 5.6: Simulation results for the Myriad-$\mathcal{CI}$MSPRT in a shift-in-mean test. Due to the symmetry, equal results are obtained for $\mathcal{H}_0/\mathcal{H}_3$ and $\mathcal{H}_1/\mathcal{H}_2$.

Shift-in-Variance Test

In the shift-in-variance test, the LFD-$\mathcal{CI}$MSPRT and the M-$\mathcal{CI}$MSPRT with Huber's score function and $c_{\text{Hub}} = 1.8$ are compared. The results are shown in Figure 5.7 and Figure 5.8, respectively.

The LFD-$\mathcal{CI}$MSPRT delivers accurate detection results up to a contamination of 10 % irrespective of network size, connectivity, or underlying hypothesis. In the case where $\mathcal{H}_3$ is true—i.e., the signal with the largest variance is active—even 20 % outliers can be tolerated. Again, the strong influence of connectivity on the runlength is visible, while the network size only has a marginal effect.

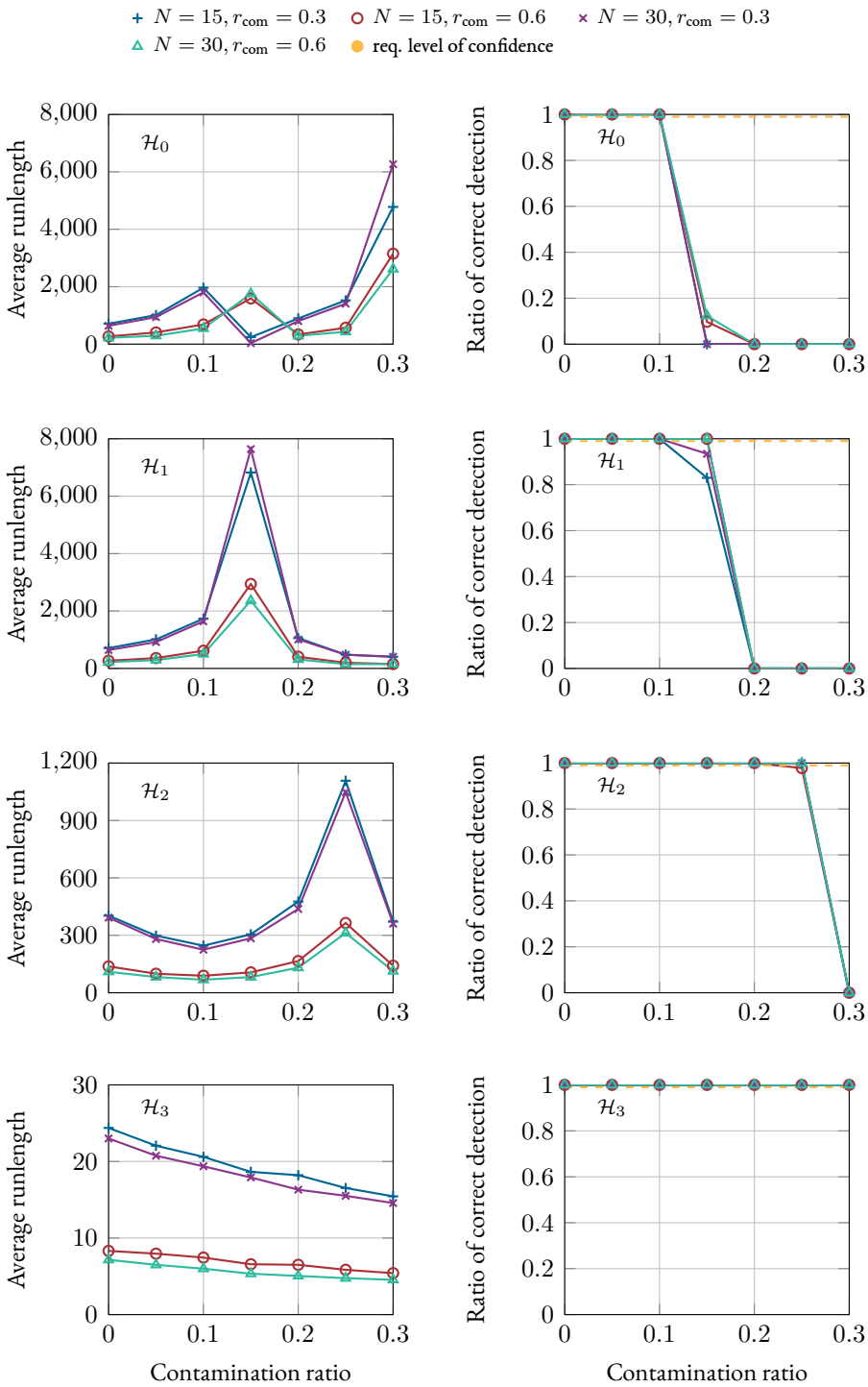

Figure 5.7: Simulation results for the LFD-$\mathcal{CI}$MSPRT in a shift-in-variance test.

Both algorithms perform similarly with the tolerable amount of contamination depending on the hypothesis—i.e., on the assumed nominal variance. For the M-$\mathcal{CI}$MSPRT, an additional dependency of the network properties on the detection performance can be observed with the connectivity having a larger impact than the network size. Under $\mathcal{H}_0$, $\mathcal{H}_1$, and $\mathcal{H}_2$, the ARL of both algorithms increases with contamination. $\mathcal{H}_3$ is the easiest test case, where outliers help to make a correct decision. Hence, the ARL decreases with increasing contamination. As before, the worst-case-optimized LFD-$\mathcal{CI}$MSPRT exhibits a considerably higher ARL overall than the M-$\mathcal{CI}$MSPRT.

5.7 Summary

In this chapter, the $\mathcal{CI}$MSPRT was proposed as a sequential detector for testing multiple hypotheses in a distributed sensor network. Simulations were run to evaluate its performance in a shift-in-mean and a shift-in-variance test. Moreover, it was verified that the algorithm's runlength can be accurately predicted beforehand.

Furthermore, five robust versions of the $\mathcal{CI}$MSPRT were developed based on LFDs and robust estimators. The effectiveness of both robustification paradigms was confirmed in a shift-in-mean and a shift-in-variance scenario. In the former case, the LFD-$\mathcal{CI}$MSPRT was shown to outperform the robust-estimator-based detectors at the cost of a considerable increase in ARL due to the worst-case optimization. The network density was shown to have a substantial impact on the ARL while the effect of the network size is negligible. Hence, the performance of the proposed algorithms scales well with the network size. In the latter case, both robustification paradigms yield a comparable detection performance on average. In the robust-estimator-based approach, however, not only the ARL but also the ratio of correct detection is influenced by the network properties. Due to this fact, a favorable choice of parameters can lead to the respective algorithm outperforming the LFD-$\mathcal{CI}$MSPRT.

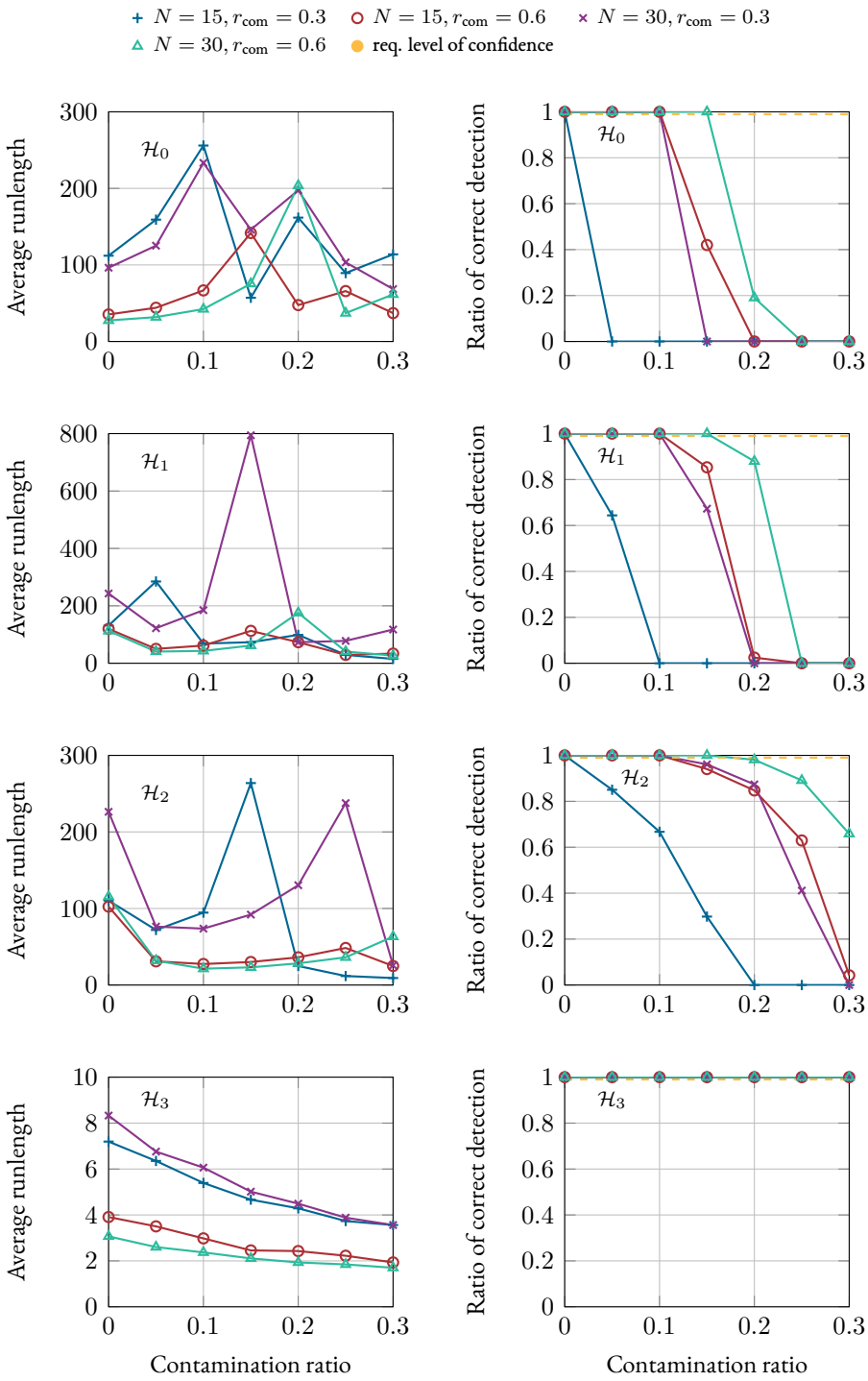

Figure 5.8: Simulation results for the M-$\mathcal{CI}$MSPRT in a shift-in-variance test.

6

Evidence-Based Sequential Hypothesis Testing

Hypothesis tests often have to satisfy several, sometimes contradicting performance criteria: Sequential methods are attractive because they can significantly reduce the amount of data needed to make a reliable decision. A distributed testing architecture is desirable due to the inherent scalability and fault tolerance of sensor networks. Finally, deviations from the Gaussian noise model arising in real-life applications make the design of robust algorithms a crucial task.

The Dempster-Shafer Theory of Evidence (DST) provides a well-established and powerful framework that offers a systematic way of dealing with these different aspects of statistical decision making [Shafer, 1976]. This chapter applies the concept of evidential probability theory based on the DST to sequential binary hypothesis testing for the first time. To this end, the same problem as in Section 4.1 is considered. Section 6.1 explains the DST by means of an example and reviews the underlying mathematics. In Section 6.2, a distributed sequential detection algorithm based on the DST is proposed. Furthermore, a novel approach for performing the basic probability assignment (BPA)—an integral part of the DST—is developed,

which is suitable for both shift-in-mean and shift-in-variance tests. Moreover, the proposed detector is robustified against distributional uncertainties. The simulations in Section 6.3 verify and evaluate the performance of the algorithm in Gaussian and non-Gaussian noise using three different BPA methods.

The contributions presented in this chapter have been published in [Leonard et al., 2018a].

6.1 The Dempster-Shafer Theory of Evidence

The Dempster-Shafer Theory of Evidence (DST) is a generalization of the Bayesian probability theory. Instead of considering probabilities for each possible event or question of interest, the DST makes a statement about the epistemic probability or *belief* associated with an event based on the beliefs for a related problem. If and to what degree these beliefs share the mathematical properties of probabilities depends on the nature of the related problem. The DST, furthermore, provides a way to combine the beliefs from multiple independent sources by means of Dempster's rule of combination [Shafer, 1976]. The concept is best described by an example.

Example Say you—or your wife—are pregnant and you are interested in the sex of the baby. After doing an ultrasound, the gynecologist is sure that you are having a boy. You belief her 75 %. Contrary to Bayesian probability theory, the remaining 25 % are not automatically attributed to you having a girl. They simply represent the possibility that the doctor is unreliable, and could point to either sex. You ask your midwife for a second opinion. Based on the pointed form of the belly, she is sure that it is a boy. Since her assessment is not scientific, you only attribute 60 % belief to it. The question of the sex of the baby now boils down to the question of how reliable the gynecologist and the midwife are. The following table shows how the evidence is combined by multiplying the beliefs.

	gynecologist reliable (75 %)	gynecologist unreliable (25 %)
midwife reliable (60 %)	boy (45 %)	boy (15 %)
midwife unreliable (40 %)	boy (30 %)	unsure (10 %)

Three of four possible events support the statement that you are having a boy. Adding the individual beliefs gives you a total belief of 90 %.

Now say that the midwife changes her mind. This changes the events as follows.

	gynecologist reliable (75 %)	gynecologist unreliable (25 %)
midwife reliable (60 %)	impossible (45 %)	girl (15 %)
midwife unreliable (40 %)	boy (30 %)	unsure (10 %)

It is impossible that both the gynecologist and the midwife are right. Hence, the possible events have to be rescaled accordingly. This yields the following assignment.

	gynecologist reliable (75 %)	gynecologist unreliable (25 %)
midwife reliable (60 %)	-	girl (27 %)
midwife unreliable (40 %)	boy (55 %)	unsure (18 %)

Thus, you are now 55 % sure your baby is a boy.

The two main ideas of the DST—namely, the extraction of beliefs via association to a related problem, and the combination of beliefs from multiple independent sources—make it a perfect fit for many modern signal processing applications such as smart homes or intelligent traffic control. Here, the task is to make a statement about a complex event based on the data from multiple sensors of different kind. Each kind of sensor can only provide data for a related subproblem. Furthermore, some sensors might be more reliable than others depending (a) on how close the subproblem is related to the event, and (b) on possible disturbances and the robustness of the sensors.

The DST is formally defined as follows. Let Ω be the so-called *frame of discernment*—i.e., the set of hypotheses under test—with the corresponding *power set* 2^Ω. The latter contains all possible subsets of Ω—i.e., all possible test outcomes including the empty set $\emptyset$, which represents the case where the available information is conflicting, and Ω, which represents the remaining uncertainty between the hypotheses. Every element of the power set is assigned a probability mass according to the BPA

$$m : 2^\Omega \to [0,1] \quad \text{such that} \quad m(\emptyset) = 0, \sum_{A \subset 2^\Omega} m(A) = 1. \tag{6.1}$$

The quantity $m(A)$ measures the belief that is committed exactly to outcome A. A belief function $Bel\ : 2^\Omega \to [0,1]$ measures the total belief committed to A by summing over all

proper subsets B of A according to

$$Bel(A) = \sum_{B \subset A} m(B).$$

This gives a lower bound on the evidence pointing to A. Obviously, if A is not further divisible into subsets, it holds that $Bel(A) = m(A)$. In that case, the terms belief and BPA are used interchangeably. An upper bound on the evidence pointing to A is given by the plausibility function $Pl : 2^{\Omega} \to [0, 1]$ with

$$Pl(A) = \sum_{A \cap B \neq \emptyset} m(B).$$

Pl sums over all sets that intersect with A, i.e., it contains all evidence that *could* commit to A. It holds that $0 \leq Bel \leq Pl \leq 1$.

Beliefs from two independent sources with belief functions Bel_1, Bel_2 and respective BPAs m_1, m_2 can be combined as long as they are non-conflicting and their so-called *focal elements* $A_1, \ldots, A_k$ and $B_1, \ldots, B_l$ intersect—i.e., as long as the possible test outcomes overlap. This can be formulated as

$$K = \sum_{\substack{i,j \\ A_i \cap B_j = \emptyset}} m_1(A_i) m_2(B_j) < 1.$$

The beliefs are combined using Dempster's rule of combination, which forms their *orthogonal sum* according to

$$\begin{aligned} Bel(A) &= Bel_1 \oplus Bel_2, \\ &= (1 - K)^{-1} \sum_{\substack{i,j \\ A_i \cap B_j = A}} m_1(A_i) m_2(B_j). \end{aligned}$$

The resulting combined belief is the sum of the belief masses of all intersecting focal elements, normalized by the summed masses of the non-intersecting—i.e., conflicting—elements. The beliefs of N independent, non-conflicting sources can be combined by applying Dempster's

rule of combination iteratively, i.e.,

$$Bel(A) = ((Bel_1 \oplus Bel_2) \oplus \cdots) \oplus Bel_N.$$

6.2 Evidence-Based Distributed Sequential Detection

In distributed sequential detection, each node k performs a sequential hypothesis test to decide between the null hypothesis $\mathcal{H}_0$ and the alternative $\mathcal{H}_1$. This yields the frame of discernment $\Omega = \{\mathcal{H}_0, \mathcal{H}_1\}$ with corresponding power set $2^\Omega = \{\emptyset, \mathcal{H}_0, \mathcal{H}_1, \Omega\}$. In the following, three different methods for performing the BPA are presented. Furthermore, it is shown how the test can be formulated using Dempster's rule of combination.

6.2.1 Basic Probability Assignment

A straightforward way to perform the BPA in a distributed, binary hypothesis test is to use the PDF of the measurement at each node [Qihang et al., 2006]. In a Gaussian environment this yields

$$\begin{aligned} m_k(t, \mathcal{H}_m) &= \frac{1}{\sqrt{2\pi}\sigma_m} e^{-\frac{(y_k(t)-\mu_m)^2}{2\sigma_m^2}}, \qquad m \in \{0, 1\}, \\ m_k(t, \Omega) &= 1 - m_k(t, \mathcal{H}_0) - m_k(t, \mathcal{H}_1), \end{aligned} \tag{6.2}$$

with μ_m and σ_m denoting the mean and the standard deviation under $\mathcal{H}_m$. A graphical representation of the BPA-PDF is shown in the first row of Figure 6.1. The left column considers a shift-in-mean scenario while the right one is concerned with a shift-in-variance test. The BPA-PDF is not a good choice as most of the mass is assigned to the uncertain event Ω where either hypothesis could be true. However, large negative (positive) measurements should lead to a decision in favor of $\mathcal{H}_0$ ($\mathcal{H}_1$) in the shift-in-mean case and always in favor of $\mathcal{H}_1$ in the shift-in-variance case. Furthermore, the lowest value of $m(\Omega)$ is reached at $y_k(t) = 0$. This is counter-intuitive. The detector should be most uncertain about its decision when both hypotheses are equally likely to be true, as it is the case in the shift-in-mean scenario.

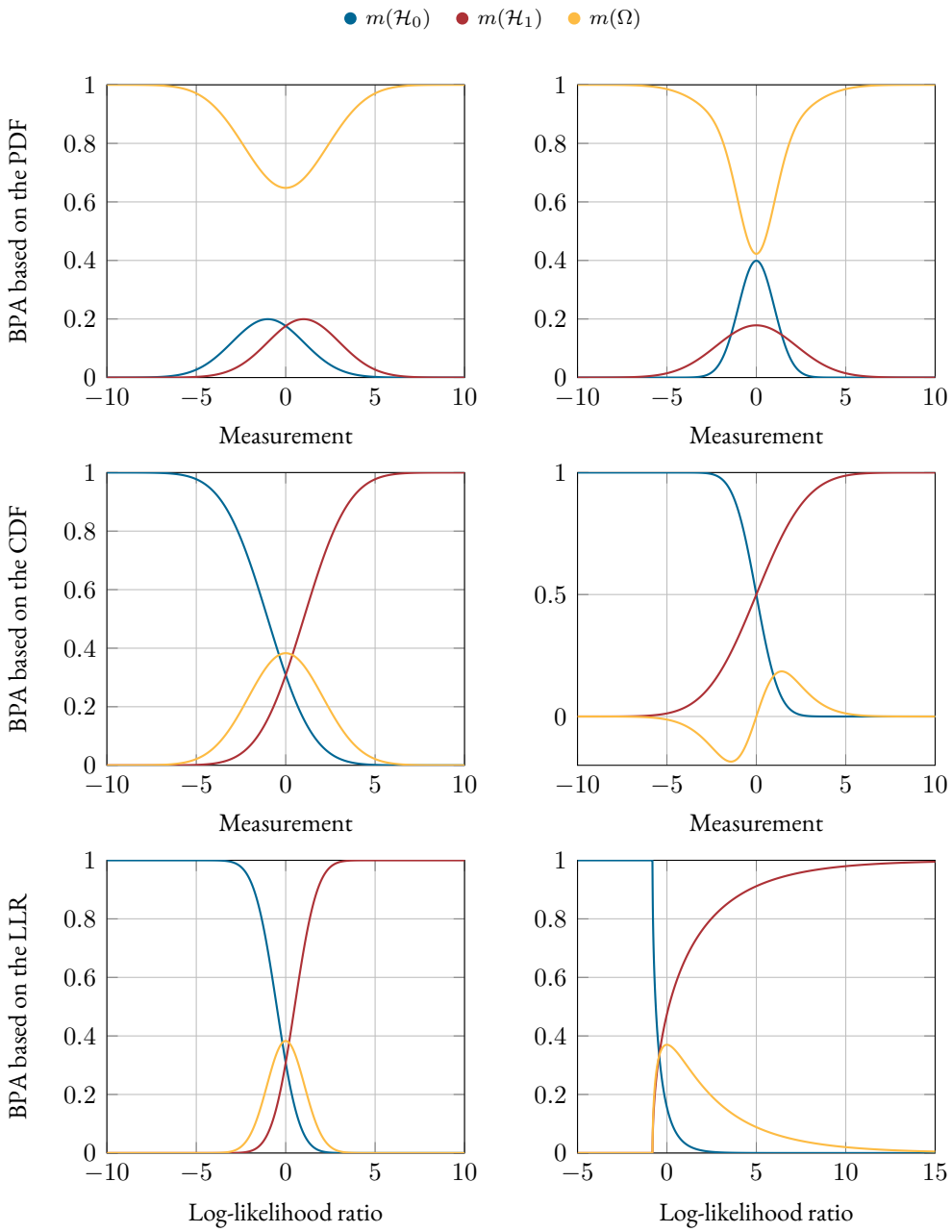

Figure 6.1: BPA for the power set $2^{\Omega} = \{\emptyset, \mathcal{H}_0, \mathcal{H}_1, \Omega\}$ in a shift-in-mean test (left column) and a shift-in-variance test (right column) based on the PDF (first row), the cumulative distribution function (CDF) (second row) and the CDF of the LLR (third row).

In [Nguyen-Thanh & Koo, 2009], a BPA based on the CDF of each node's measurement is proposed according to

$$\begin{aligned} m_k(t,\mathcal{H}_0) &= 1 - \int_{-\infty}^{y_k(t)} \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(y-\mu_0)^2}{2\sigma_0^2}} \mathrm{d}y, \\ m_k(t,\mathcal{H}_1) &= \int_{-\infty}^{y_k(t)} \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(y-\mu_1)^2}{2\sigma_1^2}} \mathrm{d}y, \\ m_k(t,\Omega) &= 1 - m_k(t,\mathcal{H}_0) - m_k(t,\mathcal{H}_1). \end{aligned} \tag{6.3}$$

A graphical representation of the BPA-CDF is given in the second row of Figure 6.1. This BPA has two main advantages over the BPA-PDF. First, $m_k(\mathcal{H}_0)$ and $m_k(\mathcal{H}_1)$ are monotone functions. This means that the assignment is unique for each $y_k(t)$. Second, in shift-in-mean tests, they are complementary due to the symmetry of the problem. This leads to an intuitive assignment of uncertainty, which is highest at the intersection. In shift-in-variance tests, however, the BPA-CDF results in negative $m_k(t,\Omega)$ since the masses do not sum up to one. Hence, (6.1) is violated and the BPA-CDF is not a valid BPA in this context.

In this work, a novel way to perform the BPA based on the CDF of the LLR of the measurement is proposed. In the shift-in-mean case, the LLR is Gaussian-distributed and the BPA-LLR is performed according to

$$\begin{aligned} m_k(t,\mathcal{H}_0) &= 1 - \int_{-\infty}^{\eta_k(t)} \frac{1}{\sqrt{2\pi}\sigma_{\eta,0}} e^{-\frac{(y-\mu_{\eta,0})^2}{2\sigma_{\eta,0}^2}} \mathrm{d}y, \\ m_k(t,\mathcal{H}_1) &= \int_{-\infty}^{\eta_k(t)} \frac{1}{\sqrt{2\pi}\sigma_{\eta,1}} e^{-\frac{(y-\mu_{\eta,1})^2}{2\sigma_{\eta,1}^2}} \mathrm{d}y, \\ m_k(t,\Omega) &= 1 - m_k(t,\mathcal{H}_0) - m_k(t,\mathcal{H}_1), \end{aligned} \tag{6.4}$$

where $\mu_{\eta,m}$ and $\sigma_{\eta,m}$ denote the mean and standard deviation of the LLR under $\mathcal{H}_m$. Details on how to derive these quantities can be found in Appendix A.1. In the shift-in-variance case,

the LLR follows a scaled and shifted χ_1^2 distribution. Thus,

$$
\begin{aligned}
m_k(t, \mathcal{H}_0) &= 1 - \int_{-\infty}^{\eta_k(t)} \frac{1}{b\sigma_0^2} f_{\chi_1^2}\left(\frac{\eta - a}{b\sigma_0^2}\right) \mathrm{d}\eta, \\
m_k(t, \mathcal{H}_1) &= \int_{-\infty}^{\eta_k(t)} \frac{1}{b\sigma_1^2} f_{\chi_1^2}\left(\frac{\eta - a}{b\sigma_1^2}\right) \mathrm{d}\eta, \\
m_k(t, \Omega) &= 1 - m_k(t, \mathcal{H}_0) - m_k(t, \mathcal{H}_1),
\end{aligned}
$$

where

$$
\begin{aligned}
a &= \log\left(\frac{\sigma_0}{\sigma_1}\right), \\
b &= \frac{1}{2}\left(\frac{1}{\sigma_0^2} - \frac{1}{\sigma_1^2}\right),
\end{aligned}
$$

and $f_{\chi_1^2}(\cdot)$ denotes the PDF of the chi-squared distribution with one degree of freedom.

The BPA-LLR is illustrated in the third row of Figure 6.1. As visible in the left column, the BPA-LLR has the same advantages as the BPA-CDF. It even leads to steeper BPA curves and a narrower Ω region. It is also applicable to shift-in-variance tests as apparent from the figure in the right column. Both $m_k(\mathcal{H}_0)$ and $m_k(\mathcal{H}_1)$ are monotone and the uncertainty is highest at $\eta_k(t) = 0$. These properties make the BPA-LLR most suitable for the considered applications.

6.2.2 Sequential Detection in a Distributed Sensor Network

At each time instant t, every node k takes a measurement and performs a BPA according to one of the methods from Section 6.2.1. Afterwards, it transmits its belief masses to its neighbors. In a next step, every node k combines the beliefs of its closed neighborhood $\mathcal{N}_k$ as well as the combined belief from the previous time step using Dempster's rule of combination according to

$$
\begin{aligned}
Bel_{\text{comb}}(\mathcal{H}_0, t) &= ((Bel_k(\mathcal{H}_0) \oplus Bel_l(\mathcal{H}_0)) \oplus \cdots) \oplus m_{\text{comb}}(\mathcal{H}_0, t-1), \quad l \in \mathcal{N}_k, \\
Bel_{\text{comb}}(\mathcal{H}_1, t) &= ((Bel_k(\mathcal{H}_1) \oplus Bel_l(\mathcal{H}_1)) \oplus \cdots) \oplus m_{\text{comb}}(\mathcal{H}_1, t-1), \quad l \in \mathcal{N}_k, \\
Bel_{\text{comb}}(\Omega, t) &= 1 - Bel_{\text{comb}}(\mathcal{H}_0, t) - Bel_{\text{comb}}(\mathcal{H}_1, t).
\end{aligned}
$$

A decision is made in favor of one of the hypotheses as soon as one of the corresponding combined beliefs reaches the predefined level of confidence $1-\alpha$. The corresponding decision and stopping rule reads as

$$\delta(t) = \begin{cases} 0, & Bel_{\text{comb}}(\mathcal{H}_0, t) \geq 1-\alpha, \\ 1, & Bel_{\text{comb}}(\mathcal{H}_1, t) \geq 1-\alpha, \\ \text{continue sampling}, & \text{otherwise.} \end{cases} \tag{6.5}$$

6.2.3 Robustifying the Detector

The DST-based sequential detector is robustified using the reliability measure introduced in [Han et al., 2012; Wang et al., 2014]. To this end, each node k determines the *max-min similarity degree* of its own belief compared to its neighborhood beliefs according to

$$\varsigma_{k,l} = \frac{\sum_{A \subset 2^\Omega} \min(m_k(A), m_l(A))}{\sum_{A \subset 2^\Omega} \max(m_k(A), m_l(A))},$$
$$\varsigma_k = \sum_{l \in \mathcal{N}_k^o} \varsigma_{k,l}.$$

Note that the sum is taken over the open neighborhood $\mathcal{N}_k^o$ as defined in Section 2.2. In words, each node k computes for every neighbor l the quotient between the minimum belief summed over all possible events, and the maximum belief summed over all events. This gives a measure for how the beliefs of node k and l deviate. By summing over the open neighborhood, a measure for the similarity of the beliefs of node k compared to its neighbors is obtained.

The similarity degrees ς_k of all nodes are collected in vector $\boldsymbol{\varsigma}$ and the reliability ρ_k of each node's belief is calculated as

$$\rho_k = \frac{\varsigma_k}{\max(\boldsymbol{\varsigma})}.$$

When combining the beliefs of its neighborhood, node k updates the belief masses of each

neighbor $l \in \mathcal{N}_k$ according to

$$\begin{aligned} m_l^*(t, \mathcal{H}_0) &= \rho_l \cdot m_l(t, \mathcal{H}_0), \\ m_l^*(t, \mathcal{H}_1) &= \rho_l \cdot m_l(t, \mathcal{H}_1), \\ m_l^*(t, \Omega) &= 1 - m_l^*(t, \mathcal{H}_0) - m_l^*(t, \mathcal{H}_1), \end{aligned} \tag{6.6}$$

before applying Dempster's rule of combination. That way, the influence of unreliable or disturbed neighbors on the decision of node k is downweighted.

6.3 Simulations

In this section, the performance of the proposed evidence-based sequential detector is evaluated in a Gaussian and a non-Gaussian environment with $\epsilon = 0.1$ and $\kappa = 10$. To this end, the shift-in-mean and the shift-in-variance test scenarios from Section 4.1 are considered with $\mu_0 = -1$, $\mu_1 = 1$, and $\sigma^2 = 2$, as well as $\sigma_n^2 = 1$ and $\sigma_x^2 = 4$, respectively. The BPA is performed based on the PDF (BPA-PDF), the CDF (BPA-CDF), and the LLR (BPA-LLR) as proposed in Section 6.2.1. Furthermore, the tests are run with and without applying the reliability measure from Section 6.2.3. A network of $N = 20$ nodes with uniformly distributed x- and y-coordinates on the interval $[0, 1]$ is used. Nodes within a radius of $r_{\text{com}} = 0.6$ are neighbors. The required confidence level $1 - \alpha$ ranges from 0.7 to 0.99. The ARL and the ratio of correct detection serve as performance metrics. The results are averaged over 10,000 Monte Carlo runs.

6.3.1 Simulation Results

The simulation results are depicted in Figure 6.2. The first row shows the results for the shift-in-mean test, which are equal under both hypotheses due to the symmetry of the problem. The second and third row show the results for the shift-in-variance test under $\mathcal{H}_0$ and $\mathcal{H}_1$, respectively. The algorithm using the BPA-PDF exhibits the largest ARL but achieves an almost perfect detection result in the shift-in-mean case and in the shift-in-variance case under $\mathcal{H}_0$. Under $\mathcal{H}_1$ the algorithm breaks down since the BPA-PDF always favors $\mathcal{H}_1$ (see Figure 6.1). Using the BPA-CDF leads to a smaller ARL with a very high ratio of correct detection in the

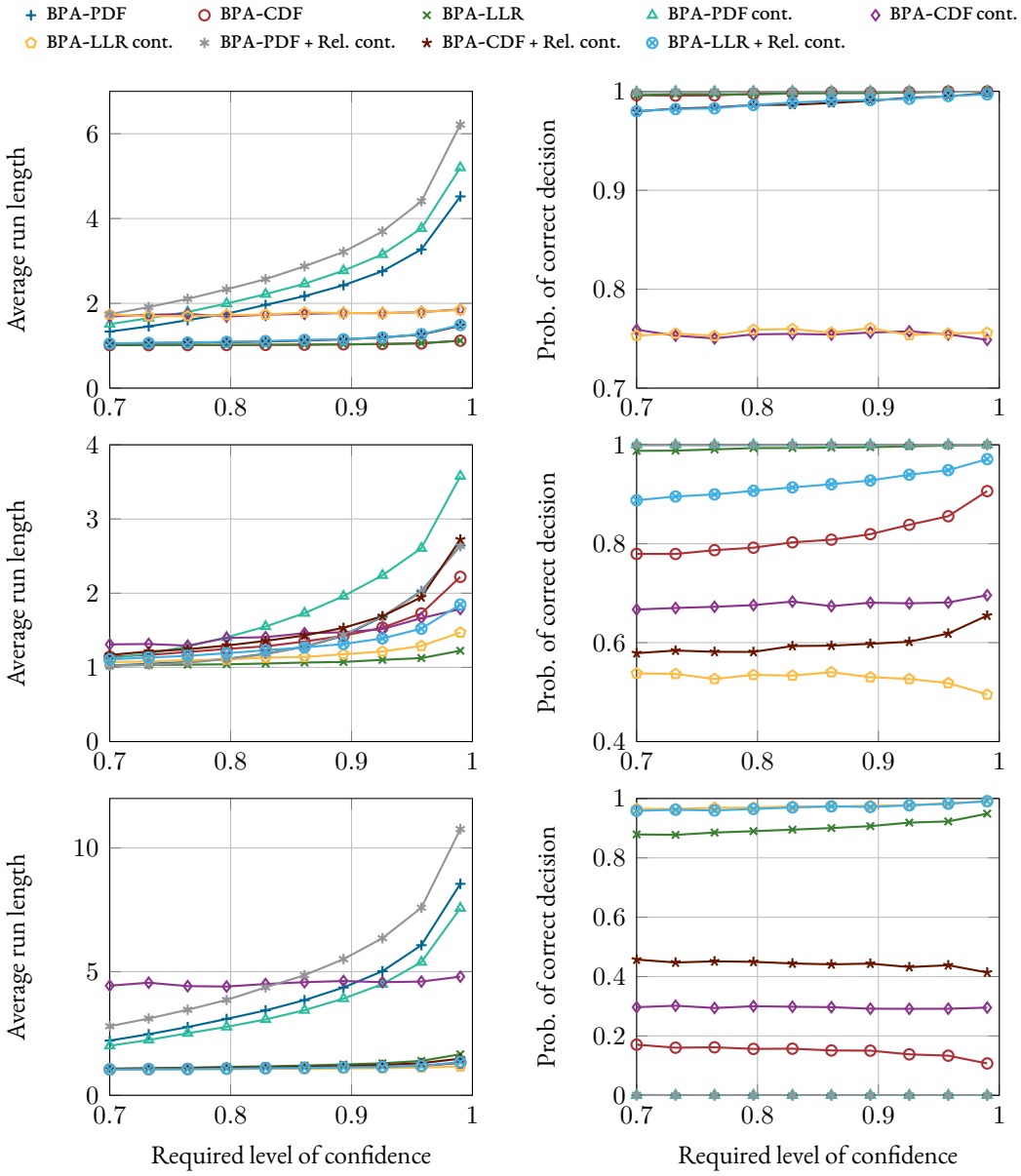

Figure 6.2: Simulation results of a shift-in-mean test (first row) and a shift-in-variance test under $\mathcal{H}_0$ (second row) and $\mathcal{H}_1$ (third row). We consider the BPA based on the PDF (BPA-PDF), on the CDF (BPA-CDF), and on the CDF of the LLR (BPA-LLR). The detector is tested with and without reliability (Rel.) The tests are performed under Gaussian noise and under contaminated noise (cont.).

shift-in-mean case. The performance degrades in the shift-in-variance case since the beliefs do not sum up to one.

The proposed BPA-LLR delivers the best results. It always exhibits a very small ARL. In a Gaussian environment, the detection result is almost perfect in the shift-in-mean case as well as in the shift-in-variance case under $\mathcal{H}_0$. Under $\mathcal{H}_1$, it is still between 0.9 and 1. In the face of outliers, the performance drops in the first two scenarios but can be restored using the reliability measure. In the shift-in-variance case under $\mathcal{H}_1$, the reliability measure is not needed because outliers help to make a correct decision.

6.4 Summary

In this chapter, a distributed sequential detector based on the DST was proposed and its performance was validated in a shift-in-mean as well as a shift-in-variance scenario. Furthermore, a novel form of BPA based on the CDF of the LLR was introduced, which works in both scenarios and outperforms existing BPA methods. Finally, the concept of reliability was successfully applied to robustify the proposed detector against distributional uncertainties of the ε-contamination type.

7

Conclusions & Outlook

The first part of this dissertation was concerned with robust sequential detection in distributed sensor networks. First, a general framework for distributed sequential detection in binary hypothesis test was developed. The framework was robustified using two different robustification paradigms—one optimized for the worst case, the other dependent on the network connectivity. While both paradigms were shown to produce reliable detectors, the question of which detector is most suited depends on the application at hand. Second, the framework was extended to multiple hypothesis testing by performing an acceptance test of hypothesis pairs. It was robustfied using the same paradigms as in the binary case and the resulting detectors were shown to work reliably. Finally, the sequential binary hypothesis test was reformulated based on evidence theory and a reliable solution for distributed sequential detection in this setup was derived.

The problems listed below are possible extensions of this work. Some of them are the topic of ongoing research.

7.1 Arbitrary Sequential Hypothesis Tests

In Chapter 4, the $\mathcal{CI}$SPRT is formulated such that it is applicable to arbitrary binary hypothesis tests as long as the test statistic is approximately normally distributed. The same holds for the $\mathcal{CI}$MSPRT in Chapter 5. Hence, it would be interesting to examine other test scenarios beyond Gaussian shift-in-mean and shift-in-variance tests. In case the test statistic follows a different distribution, the existence of suitable transformations should be investigated.

In DST-based sequential detection (Chapter 6), the applicability to different test scenarios is dictated by the employed BPA. Therefore, further methods for performing the BPA should be investigated and—ideally—an optimal method for assigning the probability mass should be derived. Moreover, the proposed algorithm should be readily extensible to multiple hypothesis testing by performing $M(M-1)$ pairwise tests as in Chapter 5 and reformulating the stopping and decision rule (6.5) accordingly.

7.2 Analysis of the Impact of Network Properties

The simulation results in Section 4.6 and Section 5.6 showed that the performance of the proposed algorithms depends strongly on the network connectivity and weakly on the network size. A mathematical analysis of the impact of different network properties on the performance would be helpful to fully understand and possibly predict the behavior of the detectors. However, the interplay of fully distributed communication, consensus-like averaging, and robustness make this a non-trivial, potentially intractable task. In case such an analysis can be done, it would be possible to further improve the robust-estimator-based versions of the $\mathcal{CI}$SPRT and the $\mathcal{CI}$MSPRT by deriving better decision thresholds as commented on in Section 4.5.

7.3 Heterogeneous Sensor Networks

The sensor networks considered in this work are all homogeneous—i.e., all sensor nodes are of the same type. In many real-world applications such as intelligent traffic control, smart homes, or health monitoring, however, the challenge is to combine the information from different kinds of sensors. A first attempt at using the $\mathcal{CI}$SPRT—albeit the original form

that is tailored to Gaussian shift-in-mean tests—in a heterogenous network was presented in [Sahu & Kar, 2014]. However, *heterogeneous* here simply refers to defective and non-defective sensors of the same kind. Distributed sequential detection in truely heterogeneous networks is so far an untapped field of research. It is especially interesting in the context of DST-based hypothesis testing since the combination of information from different sources is one of the strengths of the DST [Shafer, 1976].

PART II

Location Estimation & Tracking

8

Multi-Target Tracking

The purpose of this chapter is to introduce the basic concepts of and issues present in multi-target tracking, thereby laying the groundwork for the following chapters. First, the problem of data association is discussed in Section 8.1. Section 8.2 presents the considered state-space and measurement models. The concept of random finite sets is reviewed in Section 8.3. Finally, a review of the PHD and the PHD Filter is given in Section 8.4 and Section 8.5, respectively.

The contributions presented here and in the following chapters have been published in [Leonard & Zoubir, 2019].

8.1 Data Association

In a general target tracking scenario involving multiple sensors, every sensor obtains a number of measurements, each of which might stem from the target or clutter in the case of single-target tracking, and from one of the targets or clutter in multi-target tracking. In order to select the right measurements to be considered by the tracking algorithm, a *validation region* or gate is set up around each predicted target state such that the target measurement falls in

it with high gate probability. If more than one measurement lies within the validation region, one faces association uncertainty in determining the measurement that belongs to the corresponding target. Measurements outside the gate are unlikely to have originated from the target and can be ignored. In multi-target tracking, intersecting validation regions and the finite resolution capability of the signal processing unit add additional layers of complexity to the association problem [Bar-Shalom et al., 2009].

If the number of targets is known, the data association problem can be solved by prepending a data association step to the tracking algorithm such as the Joint Probabilistic Data Association Filter, or the Multiple Hypothesis Tracker [Reid, 1979]. However, the resource constraints in sensor networks might pose a challenge on finding distributed implementations of these approaches [Oh et al., 2007]. In addition, both algorithms assume *a priori* knowledge of the number of targets, so that a corresponding number of trackers can be initialized. Recently, the PHD Filter [Mahler, 2003] has gained increasing attention as a more rigorous approach based on random finite sets, which is able to deal with an unknown number of targets that may vary over time. Furthermore, it simplifies the data association problem in the multi-sensor case and completely circumvents it for the case of only a single sensor since the identities of the individual targets are not required [Clark, 2006]. For these reasons, the multi-target tracking algorithms presented in this work are based on this concept.

8.2 State-Space and Measurement Model

A linear state-space model—as detailed, e.g., in [Gustafsson et al., 2002]—is considered for each target at time instant $t \geq 0$. The target state vector $\boldsymbol{s}^{\text{tgt}}(t) = [\boldsymbol{x}^{\text{tgt}}(t), \dot{\boldsymbol{x}}^{\text{tgt}}(t)]^\top$ contains the target location vector $\boldsymbol{x}^{\text{tgt}}$ as well as the velocity vector $\dot{\boldsymbol{x}}^{\text{tgt}}$. For the sake of simplicity, this work focuses on 2D-environments. The target state evolves according to the state equation

$$\boldsymbol{s}^{\text{tgt}}(t) = \boldsymbol{F}(t)\boldsymbol{s}^{\text{tgt}}(t-1) + \boldsymbol{G}(t)\boldsymbol{n}^{\text{tgt}}(t). \tag{8.1}$$

The matrices $\boldsymbol{F}$ and $\boldsymbol{G}$ as well as the vector $\boldsymbol{n}^{\text{tgt}}$ will be explained shortly. Node k obtains a measurement $\boldsymbol{z}_k$ of the target location as given by the measurement equation

$$\boldsymbol{z}_k(t) = \boldsymbol{H}_k(t)\boldsymbol{s}^{\text{tgt}}(t) + \boldsymbol{\nu}_k^{\text{tgt}}(t), \qquad k \in \mathcal{M}, \tag{8.2}$$

with

$$\mathcal{M} = \{m \in \{1, \ldots, N\} \mid \|\boldsymbol{x}_m(t) - \boldsymbol{x}^{\text{tgt}}(t)\|_2 \leq r_{\text{sen}}\},$$

denoting the set of all nodes m that are located such that the Euclidean distance between their location $\boldsymbol{x}_m$ and the target location $\boldsymbol{x}^{\text{tgt}}$ is not greater than their sensing radius r_{sen}. Note that N is the total number of nodes in the network. Furthermore, $\boldsymbol{n}^{\text{tgt}}(t) \sim \mathcal{N}(\mathbf{0}_{2,1}, \boldsymbol{Q}(t))$ and $\boldsymbol{\nu}_k^{\text{tgt}}(t) \sim \mathcal{N}(\mathbf{0}_{2,1}, \boldsymbol{R}_k(t))$ denote the state and measurement noise processes, respectively, with the zero-mean vector $\mathbf{0}_{2,1} = [0, 0]^\top$. Both noise processes are spatially and temporally white, as well as uncorrelated with the initial target state $\boldsymbol{s}^{\text{tgt}}(0)$ and each other for all t. For the sake of simplicity, a time-invariant measurement noise covariance matrix is chosen as

$$\boldsymbol{R}_k(t) = \boldsymbol{R}_k = \sigma_r^2 \boldsymbol{I}_2,$$

where σ_r^2 is the variance of each component of the measurement noise and $\boldsymbol{I}_n$ denotes the identity matrix of size n.

In target tracking, the model matrices are usually chosen to be time-invariant [Gustafsson et al., 2002]. They are given by

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{I}_2 & \Delta t \boldsymbol{I}_2 \\ \mathbf{0}_{2,2} & \boldsymbol{I}_2 \end{bmatrix}, \qquad \boldsymbol{G} = \begin{bmatrix} \frac{\Delta t^2}{2} \boldsymbol{I}_2 \\ \Delta t \boldsymbol{I}_2 \end{bmatrix}, \qquad \boldsymbol{Q} = \sigma_q^2 \boldsymbol{I}_2,$$

where $\mathbf{0}_{2,2}$ is the 2×2 zero matrix. Furthermore, Δt is the time step interval in seconds with which the state-space model progresses. In addition, σ_q^2 denotes the variance of a state noise component. The sensor nodes are assumed to only obtain information on the location of a target. One common set of measurements that is often found in applications at sea is the combination of distance and bearing measurements from which an estimate of the target location can easily be calculated. Since the exact nature of the measured location information is hardly relevant for the design of a tracker, the focus is rather on how this information is processed by different tracking algorithms. Therefore, this work considers a measurement model that is based on the local target location estimates at each node and is applicable to a wide variety of applications irrespective of the exact measurement quantities. The general

measurement matrix $\boldsymbol{H}_k$ reads as

$$\boldsymbol{H}_k = \begin{bmatrix} \boldsymbol{I}_2 & \boldsymbol{0}_{2,2} \end{bmatrix}.$$

8.3 Random Finite Sets

A random finite set is an unordered set that is random in the number of its countable elements as well as in their values [Mahler, 2007; Vo, 2008; Mahler, 2014]. Therefore, this kind of set is a natural choice for representing the multi-target states and measurements in multi-target tracking: the state and measurement vectors of all targets are collected in corresponding random finite sets [Vo et al., 2003; Vo et al., 2005]. Given the realization Ξ_{t-1} of the set $\boldsymbol{\Xi}_{t-1}$ at time instant $t-1$, the multi-target state of the tracking problem can be described by the set $\boldsymbol{\Xi}_t$ according to

$$\boldsymbol{\Xi}_t = \boldsymbol{\mathcal{S}}_t(\Xi_{t-1}) \cup \boldsymbol{\mathcal{B}}_t,$$

where the survival set $\boldsymbol{\mathcal{S}}_t(\Xi_{t-1})$ denotes the random finite set of targets that already existed at time step $t-1$ and have not exited the region of interest—i.e., the region covered by the sensor network—in the transition to time step t. In addition, the birth set $\boldsymbol{\mathcal{B}}_t$ is the random finite set of new targets that spontaneously appear at the border of the region at time instant t [Mahler, 2003; Vo et al., 2005; Challa et al., 2011]. Note that the statistical behavior of $\boldsymbol{\Xi}_t$ can be described by the conditional probability $f_{t|t-1}(\Xi_t \mid \Xi_{t-1})$. Compared to the case of single-target tracking, $f_{t|t-1}(\Xi_t \mid \Xi_{t-1})$ corresponds to the state-transition PDF $f_{t|t-1}(\boldsymbol{s}^{\text{tgt}}(t) \mid \boldsymbol{s}^{\text{tgt}}(t-1))$.

The multi-target measurement model is given by the set $\boldsymbol{\Sigma}_t$ as

$$\boldsymbol{\Sigma}_t = \boldsymbol{\Theta}_t(\Xi_t) \cup \boldsymbol{\mathcal{C}}_t(\Xi_t),$$

where $\boldsymbol{\Theta}_t(\Xi_t)$ is the random finite set of measurements generated by Ξ_t. In addition, the set $\boldsymbol{\mathcal{C}}_t(\Xi_t)$ represents clutter or false alarms. Given a realization Σ_t of $\boldsymbol{\Sigma}_t$, the statistical behavior of $\boldsymbol{\Sigma}_t$ is described by the conditional probability $f_t(\Sigma_t \mid \Xi_t)$, which corresponds to the likelihood $f_t(\boldsymbol{z}(t) \mid \boldsymbol{s}^{\text{tgt}}(t))$ in single-target tracking, with $\boldsymbol{z}$ denoting the measurement vector.

8.4 The Probability Hypothesis Density

In analogy to the single-target case, the optimal Bayesian filter for multi-target tracking recursively propagates the multi-target posterior $f_{t|t}(\Xi_t \mid \Sigma_{0:t})$ over time, according to

$$f_{t|t}(\Xi_t \mid \Sigma_{0:t}) = \frac{f_t(\Sigma_t \mid \Xi_t) f_{t|t-1}(\Xi_t \mid \Sigma_{0:t-1})}{\int f_t(\Sigma_t \mid \Xi) f_{t|t-1}(\Xi \mid \Sigma_{0:t-1}) \mu_s(\mathrm{d}\Xi)},$$
$$f_{t|t-1}(\Xi_t \mid \Sigma_{0:t-1}) = \int f_{t|t-1}(\Xi_t \mid \Xi) f_{t-1|t-1}(\Xi \mid \Sigma_{0:t-1}) \mu_s(\mathrm{d}\Xi),$$

where $\Sigma_{0:t}$ denotes the set of all measurements up to time t, and μ_s is a dominating measure as described in [Vo et al., 2005]. This approach requires the evaluation of multiple integrals, which makes it even more computationally challenging than its single-target counterpart. A common solution is to find a set of statistics that yield a good approximation of the posterior, and propagate them instead [Challa et al., 2011].

In single-target tracking, the two most familiar statistics are the first- and second-order moments given by

$$\hat{\boldsymbol{s}}^{\mathrm{tgt}}(t \mid t) = \int \boldsymbol{s}^{\mathrm{tgt}}(t) \cdot f_{t|t}(\boldsymbol{s}^{\mathrm{tgt}}(t) \mid \boldsymbol{z}(t))\, \mathrm{d}\boldsymbol{s}^{\mathrm{tgt}}(t),$$

and

$$\boldsymbol{C}(t \mid t) = \int \boldsymbol{s}^{\mathrm{tgt}}(t) \boldsymbol{s}^{\mathrm{tgt}}(t)^\top \cdot f_{t|t}(\boldsymbol{s}^{\mathrm{tgt}}(t) \mid \boldsymbol{z}(t))\, \mathrm{d}\boldsymbol{s}^{\mathrm{tgt}}(t).$$

Assuming higher-order moments are negligible, the posterior $f_{t|t}(\boldsymbol{s}^{\mathrm{tgt}} \mid \boldsymbol{z}(t))$ can be approximated by a multidimensional Gaussian distribution according to

$$f_{t|t}(\boldsymbol{s}^{\mathrm{tgt}}(t) \mid \boldsymbol{z}(t)) \approx \mathcal{N}(\hat{\boldsymbol{s}}^{\mathrm{tgt}}(t \mid t), \boldsymbol{P}(t \mid t)),$$

with covariance matrix

$$\boldsymbol{P}(t \mid t) = \boldsymbol{C}(t \mid t) - \hat{\boldsymbol{s}}^{\mathrm{tgt}}(t \mid t) \hat{\boldsymbol{s}}^{\mathrm{tgt}}(t \mid t)^\top.$$

Instead of propagating the whole posterior over time, it is sufficient to propagate $\hat{\boldsymbol{s}}^{\mathrm{tgt}}(t \mid t)$ and

$\boldsymbol{P}(t \mid t)$, which leads to the well-known Kalman Filter [Kalman, 1960]. Assuming the second-order moment is negligible as well, the constant-gain Kalman Filter is obtained, which only propagates $\hat{\boldsymbol{s}}^{\text{tgt}}(t \mid t)$.

The most obvious way of extending this concept to the multi-target case is by computing the expected value $\mathbb{E}[\boldsymbol{\Xi}_t]$ as detailed in [Mahler, 2001; Mahler, 2003; Mahler, 2013]. However, the form of $f_{t|t}(\Xi_t \mid \Sigma_{0:t})$ varies with the number of targets N_{tgt} according to

$$f_{t|t}(\Xi_t \mid \Sigma_{0:t}) = \begin{cases} f_{t|t}(\emptyset \mid \Sigma_{0:t}), & \text{if } N_{\text{tgt}} = 0, \\ f_{t|t}(\{\boldsymbol{s}_1^{\text{tgt}}(t)\} \mid \Sigma_{0:t}), & \text{if } N_{\text{tgt}} = 1, \\ f_{t|t}(\{\boldsymbol{s}_1^{\text{tgt}}(t), \boldsymbol{s}_2^{\text{tgt}}(t)\} \mid \Sigma_{0:t}), & \text{if } N_{\text{tgt}} = 2, \\ \vdots & \vdots \\ f_{t|t}(\{\boldsymbol{s}_1^{\text{tgt}}(t), \ldots, \boldsymbol{s}_n^{\text{tgt}}(t)\} \mid \Sigma_{0:t}), & \text{if } N_{\text{tgt}} = n, \end{cases}$$

where $\emptyset$ denotes an empty set, n is the unknown number of targets, and $\boldsymbol{s}_n^{\text{tgt}}(t)$ is the state vector of target n. Consequently, $\mathbb{E}[\boldsymbol{\Xi}_t]$ would have to be calculated as a sum of expectations according to

$$\begin{aligned} \mathbb{E}[\boldsymbol{\Xi}_t] = \emptyset \cdot f_{t|t}(\emptyset \mid \Sigma_{0:t}) + \int \{\boldsymbol{s}_1^{\text{tgt}}(t)\} \cdot f_{t|t}(\boldsymbol{s}_1^{\text{tgt}}(t) \mid \Sigma_{0:t}) \mathrm{d}\boldsymbol{s}_1^{\text{tgt}}(t) \\ + \ldots + \int \{\boldsymbol{s}_1^{\text{tgt}}(t), \ldots, \boldsymbol{s}_n^{\text{tgt}}(t)\} \cdot f_{t|t}(\boldsymbol{s}_1^{\text{tgt}}(t), \ldots, \boldsymbol{s}_n^{\text{tgt}}(t) \mid \Sigma_{0:t}) \mathrm{d}\boldsymbol{s}_1^{\text{tgt}}(t) \ldots \mathrm{d}\boldsymbol{s}_n^{\text{tgt}}(t). \end{aligned}$$

Since addition and subtraction of finite sets are undefined, this operation is undefined as well.

In order to obtain an analogue of the expectation for the multi-target case, one needs to find a one-to-one transformation function ϕ that transforms the realization Ξ_t of the random finite set $\boldsymbol{\Xi}_t$ of multi-target states into elements $\phi(\Xi_t)$ of a vector space $\mathcal{S}$ [Mahler, 2001]. In addition, ϕ should transform set-theoretic operations into corresponding vector-algebra operations so that the union of two non-overlapping sets corresponds to the addition of their transformations. In this case, one can compute *indirect* first-order moments of the form $\mathbb{E}[\phi(\boldsymbol{\Xi}_t)]$.

A suitable choice of transformation function is presented in [Mahler, 2001] as

$$\phi(\Xi_t) = \sum_{\boldsymbol{s}_n^{\text{tgt}}(t) \in \Xi_t} \delta(\boldsymbol{s}(t) - \boldsymbol{s}_n^{\text{tgt}}(t)),$$

where $\boldsymbol{s}(t)$ is an arbitrary state vector in the vector space $\mathcal{S}$ at time step t and $\delta(\boldsymbol{s}(t) - \boldsymbol{s}_n^{\text{tgt}}(t))$ denotes Dirac's delta function centered at target state $\boldsymbol{s}_n^{\text{tgt}}(t)$. Taking the expectation leads to the so-called Probability Hypothesis Density (PHD).

The PHD $D_{t|t}(\boldsymbol{s}(t)|\Sigma_{0:t})$ is given by

$$D_{t|t}(\boldsymbol{s}(t) \mid \Sigma_{0:t}) = \sum_{\boldsymbol{s}_n^{\text{tgt}}(t) \in \Xi_t} \int \delta(\boldsymbol{s}(t) - \boldsymbol{s}_n^{\text{tgt}}(t)) f_{t|t}(\boldsymbol{s}(t) \mid \Sigma_{0:t}) \delta \boldsymbol{s}(t),$$

where $\int f(Y)\delta Y$ denotes a set integral. The PHD has the following two properties:

1. The expected number of targets $\hat{N}_{\text{tgt}}(t)$ at time step t is obtained by integrating the PHD according to

$$\hat{N}_{\text{tgt}}(t) = \int D_{t|t}(\boldsymbol{s}(t) \mid \Sigma_{0:t}) \mathrm{d}\boldsymbol{s}(t).$$

   This is in contrast to PDFs, which always integrate to 1.

2. Estimates of the individual target states can be found by searching for the $\left\lfloor \hat{N}_{\text{tgt}} \right\rceil$ highest peaks of the PHD, where $\lfloor \cdot \rceil$ denotes rounding to the nearest integer.

Because of these two properties, the number of targets as well as their states can be estimated independently at each time step without any knowledge of their identities. That way, the data association issue is avoided. However, this also means that PHD filters cannot deliver the continuous track of a specific target. If continuous tracks are required, an additional association step has to be performed. Two possible association algorithms for track continuity can be found in [Clark, 2006].

8.5 The Probability Hypothesis Density Filter

The PHD Filter is an approach for recursively propagating the PHD $D_{t|t}(\boldsymbol{s}(t) \mid \Sigma_{0:t})$ at time step t given measurements up to time step t over time. If the random finite set $\boldsymbol{\Xi}$ is Poisson-

distributed, then its PHD is equal to its intensity function and, hence, a sufficient statistic [Mahler, 2003]. In this case, the PHD recursion is given by the following prediction and update equations:

$$D_{t|t-1}(\boldsymbol{s}(t) \mid \Sigma_{0:t-1}) = b_t(\boldsymbol{s}(t)) + \int p_{\mathrm{S}}(\boldsymbol{s}(t-1)) f_{t|t-1}(\boldsymbol{s}(t) \mid \boldsymbol{s}(t-1)) \times D_{t-1|t-1}(\boldsymbol{s}(t-1) \mid \Sigma_{0:t-1}) \mathrm{d}\boldsymbol{s}(t-1) \tag{8.3}$$

$$D_{t|t}(\boldsymbol{s}(t) \mid \Sigma_{0:t}) = \left[1 - p_{\mathrm{D}} + \sum_{\boldsymbol{z} \in \Sigma_t} \frac{p_{\mathrm{D}} f_t(\boldsymbol{z} \mid \boldsymbol{s}(t))}{\lambda_{\mathrm{FA}} c_{\mathrm{FA}}(\boldsymbol{z}) + p_{\mathrm{D}} \int f_t(\boldsymbol{z} \mid \boldsymbol{s}(t)) D_{t|t-1}(\boldsymbol{s}(t) \mid \Sigma_{0:t-1}) \mathrm{d}\boldsymbol{s}(t)}\right] D_{t|t-1}(\boldsymbol{s}(t) \mid \Sigma_{0:t-1}) \tag{8.4}$$

Note that $b_t(\boldsymbol{s}(t))$ is the PHD of the birth set $\boldsymbol{\mathcal{B}}_t$ of new targets appearing at time step t. In addition, $p_{\mathrm{S}}(\boldsymbol{s}(t-1))$ denotes the probability that a target survives the transition from time step $t-1$ to t. The probability of survival depends on the previous state $\boldsymbol{s}(t-1)$ because a target that is close to the border of the region of interest and has a velocity vector pointing away from it is unlikely to be present at time step t. Furthermore, $f_{t|t-1}(\boldsymbol{s}(t) \mid \boldsymbol{s}(t-1))$ and $f_t(\boldsymbol{z} \mid \boldsymbol{s}(t))$ denote the transition probability and the likelihood, respectively. The probability of detection p_{D} is constant over time and the tracker's field of view since it is assumed that all targets can be detected if the region is covered. The term $\lambda_{\mathrm{FA}} c_{\mathrm{FA}}(\boldsymbol{z})$ represents Poisson-distributed false-alarms due to clutter, where λ_{FA} is the false alarm parameter, which is distributed according to its spatial distribution $c_{\mathrm{FA}}(\boldsymbol{z})$.

9

Distributed Multi-Target Tracking

This work considers distributed multi-target tracking in a sensor network with 1-coverage of the region of interest, i.e., the sensor nodes have non- or barely overlapping fields of view and are distributed such that maximum area coverage is attained [Wang et al., 2003]. An exemplary network layout with these properties is depicted in Figure 9.1. Autonomous distribution algorithms for realizing such a topology have been studied in [Balthasar et al., 2014]. The nodes in the network communicate with their neighbors in order to collaboratively detect and track targets in the considered region. In addition, all of the sensors are equipped with a signal processing unit, allowing them to form decisions without a fusion center. That way, the network can autonomously react to events such as the detection of an intruder without relying on a network operator. For the sake of simplicity, the network is considered to be static. However, the consideration of mobile sensor nodes would enable reactions such as target pursuit or escape.

Since the field of view and communication radius of each node are limited, a target is only seen by a subset of the network, which changes as the target moves. Hence, at each time instant, there is an *active* and an *inactive* part of the network. The goal, thus, is to detect and observe the target in a distributed and collaborative fashion as it travels across the region

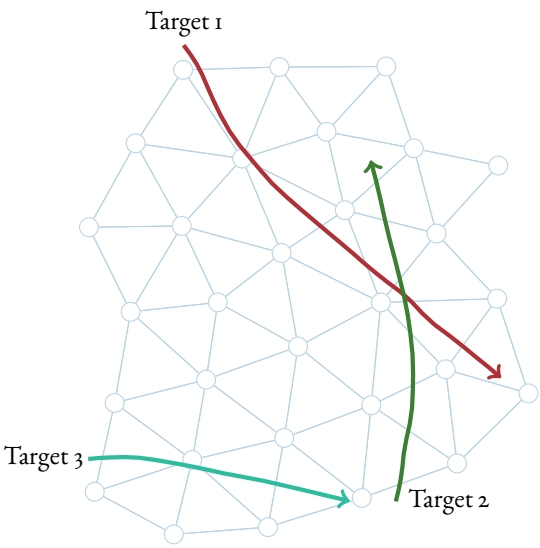

Figure 9.1: Distributed sensor network with 1-coverage of the region of interest and three exemplary target tracks.

of interest, rather than reaching a network-wide consensus on its state and have the estimate available at each node.

In Section 9.2, the D-PPHDF is introduced as a distributed Particle Filter implementation of the PHD Filter, which uses neighborhood communication to collaboratively estimate and track a single-sensor PHD at each node in the active subnetwork. Before going into the details of the algorithm, the concept of adaptive target birth (ATB) is briefly reviewed in Section 9.1 and the modification considered in this work is discussed. Section 9.3 examines the computational complexity and communication load of the proposed algorithm. Finally, in Section 9.4, a method for robustifying the D-PPHDF using robust estimators is presented.

9.1 Adaptive Target Birth

Standard formulations of the PHD Filter consider the PHD $b_t(\boldsymbol{s}(t))$ of the birth set $\boldsymbol{\mathcal{B}}_t$ to be known *a priori* [Ristic et al., 2012]. For typical tracking applications such as air surveillance, this is a reasonable assumption since new targets should appear at the border of the region of interest given continuous observation. An alternative is to make the target birth process

adaptive and measurement-driven as suggested in [Ristic et al., 2010; Ristic et al., 2012]. To this end, the PHD—and consequently the set of particles and weights approximating it in a Particle Filter implementation—is split into two densities corresponding to persistent objects, which have survived the transition from time step $t-1$ to t, and newborn objects, respectively.

In [Ristic et al., 2010; Ristic et al., 2012], the PHD of newborn objects is approximated by randomly placing N_p new particles around each target measurement, with N_p denoting the number of particles per target. This approach can be improved by only considering measurements with no noticeable impact on any persistent particle weight, as these may indicate the appearance of a new target. That way, the number of newborn particles is further reduced and a possible overlap between persistent and newborn PHD is avoided. With the transition to time step $t+1$, the newborn particles become persistent. Furthermore, by performing the ATB step towards the end of each iteration of the algorithm and only considering the particles representing the persistent PHD in the prediction, weighting, and resampling steps, the update equation (8.4) does not have to be modified as in [Ristic et al., 2010; Ristic et al., 2012].

While ATB delays the tracking algorithm by one time step, it is much more efficient as it only places new particles in regions in which a target is likely to be found. In addition, there is no need for an explicit initialization step since the first incoming target will trigger the deployment of a newborn particle cloud around its corresponding measurement.

9.2 The Diffusion Particle PHD Filter

The proposed Diffusion Particle PHD Filter (D-PPHDF) is an extension of the single-sensor Particle PHD Filter [Vo et al., 2003; Clark, 2006; Hong et al., 2011] for the multi-sensor case. Furthermore, it relies on ATB for a more efficient target detection. The communication scheme employed to exchange measurements and estimates between nodes is inspired by the two-step communication used in the context of Diffusion Adaptation [Sayed, 2013]. However, the algorithm does not rely on least-mean-squares or any kind of adaptive filter for that matter. First, each node k in the active part of the network obtains an intermediate estimate of the states of the targets present, i.e., of the PHD of persistent targets—represented by the set $\left\{\boldsymbol{s}^p_{k,\mathrm{pers}}(t), w^p_{k,\mathrm{pers}}(t)\right\}_{p=1}^{N_{k,\mathrm{pers}}(t)}$ of persistent particles with corresponding weights—based on neighborhood measurements. In other words, every active node runs a separate

Particle PHD Filter with access to measurements from its closed neighborhood $\mathcal{N}_k$. Second, it combines the intermediate estimates from its neighborhood to a final, collaborative estimate. To this end, the persistent particle sets of all neighbors are merged into a collective set $\left\{\boldsymbol{s}^p_{k,\text{coll}}(t), w^p_{k,\text{coll}}(t)\right\}_{p=1}^{N_{k,\text{coll}}(t)}$ of persistent neighborhood particles and corresponding weights before the clustering step, with $N_{k,\text{coll}}(t)$ denoting the number of collective persistent neighborhood particles.

The individual steps of the D-PPHDF are as follows:

- Merging:
  The collective persistent neighborhood particles $\boldsymbol{s}^p_{k,\text{coll}}(t-1)$ and newborn particles $\boldsymbol{s}^p_{k,\text{new}}(t-1)$ of node k are collected in the sets $\left\{\boldsymbol{s}^p_{k,\text{coll}}(t-1), w^p_{k,\text{coll}}(t-1)\right\}_{p=1}^{N_{k,\text{coll}}(t-1)}$ and $\left\{\boldsymbol{s}^p_{k,\text{new}}(t-1), w^p_{k,\text{new}}(t-1)\right\}_{p=1}^{N_{k,\text{new}}(t-1)}$ with their respective weights $w^p_{k,\text{coll}}(t-1)$ and $w^p_{k,\text{new}}(t-1)$. These sets are merged to become $\left\{\boldsymbol{s}^p_{k,\text{tot}}(t), w^p_{k,\text{tot}}(t)\right\}_{p=1}^{N_{k,\text{tot}}(t)}$, i.e., the total set of particles and weights of node k at time step t. Here, $N_{k,\text{tot}}(t)$ is the total number of particles of node k at time step t, which is given by

  $$N_{k,\text{tot}}(t) = N_{k,\text{coll}}(t-1) + N_{k,\text{new}}(t-1),$$

  with $N_{k,\text{coll}}(t-1)$ and $N_{k,\text{new}}(t-1)$ denoting the respective number of persistent neighborhood and newborn particles at the previous time step. Note that since the sets of particles and weights represent PHDs, merging the sets corresponds to summing these PHDs.

- Predicting:
  Each particle is propagated through the system model to become a persistent particle. The system model is assumed to be the same for each target and given by (8.1). Since the process noise is captured by the spread of the particle cloud, the respective term can be removed from the equation, yielding

  $$\boldsymbol{s}^p_{k,\text{pers}}(t) = \boldsymbol{F}\boldsymbol{s}^p_{k,\text{tot}}(t), \qquad p = 1, \ldots, N_{k,\text{tot}}(t). \tag{9.1}$$

  The corresponding weights are multiplied with the probability of survival p_S, which is assumed to be constant for the sake of simplicity*, according to

  $$w^p_{k,\text{pers}}(t \mid t-1) = p_\text{S} w^p_{k,\text{tot}}(t), \qquad p = 1, \ldots, N_{k,\text{pers}}(t). \tag{9.2}$$

*A constant probability of survival p_S is a reasonable assumption if the targets move relatively slowly with respect to the observation time and the size of the region of interest.

The prediction of particles and weights corresponds to the second term in (8.3).

- Measuring & Broadcasting I:
  The sensor nodes obtain measurements of the targets and forward them to their neighbors.

- Weighting:
  The persistent particle weights of node k are updated by applying a weighting step corresponding to (8.4) iteratively for each neighbor. Using the product operator, this weighting step can be compactly denoted as

$$w^p_{k,\text{pers}}(t) = \prod_{l \in \mathcal{N}_k} \left[1 - p_\text{D} + \sum_{\boldsymbol{z}_j \in \Sigma^l_t} w^p_{k,j,\text{update}}(t) \right] w^p_{k,\text{pers}}(t \,|\, t-1), \tag{9.3}$$

with

$$w^p_{k,j,\text{update}}(t) = \frac{p_\text{D} f_t(\boldsymbol{z}_j \,|\, \boldsymbol{x}^p(t))}{\lambda_\text{FA} c_\text{FA}(\boldsymbol{z}_j) + \mathcal{L}(\boldsymbol{z}_j)}, \tag{9.4}$$

where Σ^l_t is the set of measurements obtained by node l and $\mathcal{L}(\boldsymbol{z}_j)$ is calculated as

$$\mathcal{L}(\boldsymbol{z}_j) = \sum_{q=1}^{N^p_{k,\text{pers}}(t)} p_\text{D} f_t(\boldsymbol{z}_j \,|\, \boldsymbol{x}^q(t)) w^q_{k,\text{pers}}(t \,|\, t-1). \tag{9.5}$$

Note that $f_t(\boldsymbol{z}_j \,|\, \boldsymbol{x}^p(t))$ is the likelihood and $\boldsymbol{x}^p(t)$ is the location vector of particle p.

Afterwards, each node k obtains the set $\Sigma^k_{i,\text{cand}}$ of candidate measurements—i.e., measurements that are not responsible for the highest weighting of any persistent particle—to be used in the ATB step later on. The set $\Sigma^k_{i,\text{cand}}$ is found according to

$$\Sigma^k_{i,\text{cand}} = \Sigma^k_t \backslash \left\{ \boldsymbol{z}_{m_p} \,\middle|\, m_p = \arg\max_j w^p_{k,j,\text{update}}(t), p = 1, \ldots, N_{k,\text{pers}}(t) \right\}. \tag{9.6}$$

- Resampling:
  Each node k calculates its own expected number of targets $\hat{N}_{k,\text{tgt}}(t)$ from its total per-

sistent particle mass according to

$$\hat{N}_{k,\text{tgt}}(t) = \left\lfloor \sum_{p=1}^{N_{k,\text{tot}}(t)} w^p_{k,\text{pers}}(t) \right\rceil . \tag{9.7}$$

Consequently, the number of persistent particles of node k is updated as

$$N_{k,\text{pers}}(t) = \hat{N}_{k,\text{tgt}}(t) N_\text{p}.$$

Furthermore, the set of persistent particles of node k has to be resampled by drawing $N_{k,\text{pers}}(t)$ particles with replacement from it. Note that the probability of drawing particle p is given by $\frac{w^p_{k,\text{pers}}(t)}{\hat{N}_{k,\text{tgt}}(t)}$ since the weights do not sum to unity. Then, the weights are reset to equal values as

$$w^p_{k,\text{pers}}(t) = \frac{\hat{N}_{k,\text{tgt}}(t)}{N_{k,\text{pers}}(t)}, \qquad p = 1, \ldots, N_{k,\text{pers}}(t). \tag{9.8}$$

- Broadcasting II:
  Every node k transmits $\left\{ \boldsymbol{s}^p_{k,\text{pers}}(t), w^p_{k,\text{pers}}(t) \right\}_{p=1}^{N_{k,\text{pers}}(t)}$—i.e., its set of resampled persistent particles and weights—to its neighbors.

- Clustering:
  Each node k forms a collective set of persistent neighborhood particles $\boldsymbol{s}^p_{k,\text{coll}}(t)$ and corresponding weights $w^p_{k,\text{coll}}(t)$ according to

$$\left\{ \boldsymbol{s}^p_{k,\text{coll}}(t), w^p_{k,\text{coll}}(t) \right\}_{p=1}^{N_{k,\text{coll}}(t)} = \bigcup_{l \in \mathcal{N}_k} \left\{ \boldsymbol{s}^p_{l,\text{pers}}(t), w^p_{l,\text{pers}}(t) \right\}_{p=1}^{N_{l,\text{pers}}(t)},$$

  with

$$N_{k,\text{coll}}(t) = \sum_{l \in \mathcal{N}_k} N_{l,\text{pers}}(t),$$

  denoting the number of collective persistent neighborhood particles of node k. As in the merging step, this corresponds to summing the corresponding PHDs to obtain an updated single-sensor PHD with a probability distribution reflecting the information of the entire neighborhood of node k. Note that the PHDs might not be independent if a target is detected by more than one neighbor. However, this is not a problem since

merging the particle sets simply results in the respective target being represented by more particles. Hence, node k will be able to estimate the corresponding location more accurately.

The estimated target states are found by clustering the collective persistent particles. Since the expected number of targets $\hat{N}_{l,\text{tgt}}(t),\ l \in \mathcal{N}_k$ might be different for each neighbor, *hierarchical clustering of the single-linkage type* [Everitt et al., 2011] is used. Here, the sum of the expected number of targets over the neighborhood can serve as an upper bound for the number of clusters. Note, however, that if two targets are close to each other, clustering algorithms might not be able to resolve both targets correctly.

- Roughening:
  A roughening step is performed to counter sample impoverishment [Gordon et al., 1993]. To this end, an independent jitter $\boldsymbol{s}^j(t)$ is added to every resampled particle. Each component $s_c^j(t),\ c = 1, \ldots, d$ of the jitter with dimensionality d is sampled from the Gaussian distribution $\mathcal{N}(0, (\sigma_c^j(t))^2)$. The component-wise standard deviation of the jitter is given by

$$\sigma_c^j(t) = K E_c N_{k,\text{coll}}(t)^{-1/d}, \tag{9.9}$$

  where E_c is the interval length between the maximum and minimum samples of the respective component. To avoid evaluating E_c separately for each particle cluster, it is assigned an empirically found constant value.* Note that $d = 4$ since the dimensionality of the jitter vector $\boldsymbol{s}^j(t)$ and the particle state vector $\boldsymbol{s}^p(t)$ have to coincide. In addition, K is a tuning constant, which controls the spread of the particle cloud.

- Adaptive Target Birth:
  N_p new particles are placed randomly around each candidate measurement $\boldsymbol{z}_j \in \Sigma_{i,\text{cand}}^k$ leading to a total number of $N_{k,\text{new}}(t) = N_\text{p} \cdot |\Sigma_{i,\text{cand}}^k|$ newborn particles for node k. Every newborn particle is associated with a weight that is chosen according to

$$w_{k,\text{new}}^p(t) = \frac{p_\text{B}}{N_{k,\text{new}}^p(t)}, \qquad p = 1, \ldots, N_{k,\text{new}}(t), \tag{9.10}$$

  where p_B is the probability of birth. Depending on the application, p_B can depend on time as well as on the location of the respective particle. For simplicity, the probability that a new target enters the region of interest is assumed to be equal for all locations

*Since the noise variances as well as the network topology are fixed, the true value of E_c will not change significantly over time and between clusters, so this is a valid simplification.

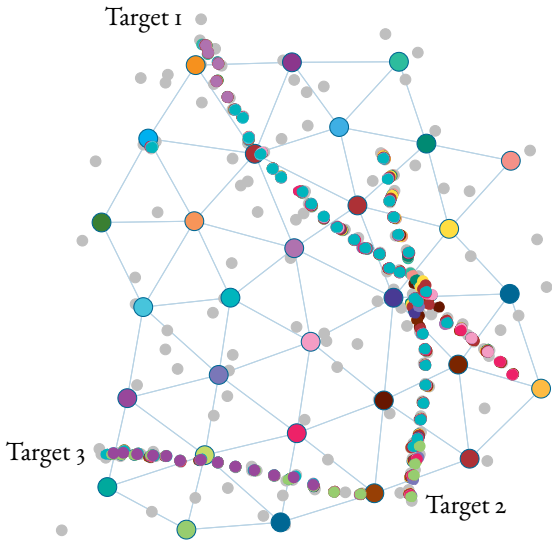

Figure 9.2: Example of tracking three targets with the D-PPHDF. The small colored dots represent the target location estimates obtained by the respective node with the same color. The light gray dots show the collective measurements obtained by all nodes in the network.

in the birth region over time. The target birth process corresponds to the first term in (8.3).

Figure 9.2 shows an example of tracking three targets, which move along the deterministic tracks depicted in Figure 9.1, using the D-PPHDF. Note that each small colored dot corresponds to a target location estimate obtained by the respective node with the same color while the light gray dots represent the collective measurements from all nodes. From this illustration, the following properties of the D-PPHDF are apparent: First, the algorithm only delivers separate location estimates—represented by the small colored dots—for each time instant rather than continuous tracks, which—as mentioned before—is a common property of PHD filters. Second, the network as a whole would be able to correctly track all three targets while a single node only obtains the locally relevant subtracks of the targets in its vicinity. Third, the employed two-step communication scheme is able to extend the vicinity of a node far beyond its own sensing radius of $r_{\text{sen}} = 6$ m in this case. This can, for instance, be seen from the fact that the light green node at the bottom left of the network is able to obtain location estimates of target 2, which enters the region of interest from the south. Finally, Figure 9.2 also illustrates the resolution problem of clustering. When targets 1 and 2, which enter the region from the north and the south, respectively, cross paths, the nodes in their vicinity see them as just one

target. This leads to an aggregation of target location estimates around the crossing point.

The pseudo-code of the D-PPHDF can be found in Appendix B.1.

9.3 Computational Complexity and Communication Load

In this section, the computational complexity and the communication load imposed by the D-PPHDF on each node in the active subnetwork is investigated. The following steps are performed at every time instant t but time dependency is omitted for simplicity. Note that each of the steps scales with the number of active nodes when considering the computational complexity of the network as a whole.

- Prediction:
  The prediction step described in (9.1) and (9.2) is performed for each particle at every active node. Hence, it scales with the number of particles $N_{k,\text{tot}}$ and the dimensionality d of the particle vectors. In order to obtain a tractable expression for the computational complexity, each node is assumed to have the same number of particles N_{tot}.
  $\Rightarrow \mathcal{O}(N_{\text{tot}} d)$

- Weighting:
  Each particle is updated in the weighting step given by (9.3)-(9.6). The weight update as well as the designation of candidate measurements for ATB depends on the neighborhood size $|\mathcal{N}_k|$ of node k, and the number of measurements $\left|\Sigma^l\right|$ of each of its neighbors l. For tractability reasons, each node is assumed to have the same number of neighbors N_{nb}, and to obtain the same number of measurements N_{meas}.
  $\Rightarrow \mathcal{O}(N_{\text{tot}} N_{\text{nb}} N_{\text{meas}})$

- Resampling:
  The estimation of the number of targets and the resampling step in (9.7) and (9.8) are linear in the number of particles used for the calculation [Gustafsson, 2010]. For the sake of simplicity, each active node is assumed to have the same estimate of the number of targets N_{tgt}.
  $\Rightarrow \mathcal{O}(N_{\text{tot}} + N_{\text{active}} N_{\text{tgt}} N_{\text{p}})$

- Clustering:
  The complexity of single-linkage clustering is cubic in the number of particles, i.e., in the number of neighbors N_{nb} of each node, the estimated number of targets N_{tgt}, the

number of particles per target N_p, and the dimensionality d of the particles [Murtagh, 1983].
$\Rightarrow \mathcal{O}((N_\mathrm{nb} N_\mathrm{tgt} N_\mathrm{p} d)^3)$

- Roughening:
  Roughening (9.9) is performed for every collective particle and is linear in the dimensionality of the particles.
  $\Rightarrow \mathcal{O}(N_\mathrm{nb} N_\mathrm{tgt} N_\mathrm{p} d)$

- Adaptive Target Birth:
  The birth process depends on the number of particles per target N_p as well as the number of candidate measurements N_cand, which is assumed to be equal for each active node to ensure tractibility.
  $\Rightarrow \mathcal{O}(N_\mathrm{p} N_\mathrm{cand})$

As far as the communication load is concerned, the D-PPHDF requires the broadcasting of measurements—i.e., two scalars per measurement—over the neighborhood in the first broadcasting step. In the second step, the sets of particles and weights—i.e., five scalars per particle—are transmitted. Clearly, the communication load strongly depends on the number of nodes in the network, or more precisely on the number of active nodes and the size of their respective neighborhood.

As an extension of the D-PPHDF, one could think of changing the second broadcasting step and transmit Gaussian Mixture Model representations—instead of the actual particles and weights—that will be resampled at the receiver node (see, e.g., [Vo & Ma, 2006]). That way, communication load could be reduced to transmitting only a few scalars in the second broadcasting step at the cost of estimation accuracy and additional computational complexity. However, a thorough treatment of this extension is beyond the scope of this work.

9.4 Robust Versions of the Diffusion Particle PHD Filter

In the first part of the thesis, two different paradigms for robustifying distributed sequential detectors were introduced, namely, LFDs—as detailed in Section 4.4—and robust estimators—as detailed in Section 4.5. As will be evident shortly, the formulation of the D-PPHDF allows for the use of the latter paradigm also in the context of multi-target tracking.

Outlying measurements can have a considerable impact on the weighting step of the D-PPHDF since this is where a decision on the importance of each individual particle is made. If the measurement set contains outliers, particles that would normally vanish in the resampling step are assigned a certain weight that causes them to be resampled and remain as viable target locations. Therefore, the weighting step is where the robustification has to take place.

The weight update $w^p_{k,j,\text{update}}(t)$ as defined in (9.4) is calculated for every measurement taken by the closed neighborhood of node k at time t. In (9.3), the sum of all of these update terms is taken. By interpreting the sum as a sample mean weighted by the number of measurements, the update equation becomes

$$w^p_{k,\text{pers}}(t) = \prod_{l \in \mathcal{N}_k} \left[1 - p_\text{D} + |\Sigma^l_t| \; \hat{w}^p_{k,\text{update}}(t) \right] w^p_{k,\text{pers}}(t \,|\, t-1), \tag{9.11}$$

with

$$\hat{w}^p_{k,\text{update}}(t) = \frac{1}{|\Sigma^l_t|} \sum_{\boldsymbol{z}_j \in \Sigma^l_t} w^p_{k,j,\text{update}}(t).$$

To robustify (9.11), the sample mean $\hat{w}^p_{k,\text{update}}$ can be replaced by a robust estimator. This work focuses on the median—as given in Section 2.1.2—and the M-estimator with Huber's score function—as detailed in Section 2.1.3. The two resulting robust distributed tracking algorithms will be referred to as Median-D-PPHDF and M-D-PPHDF, respectively. As in the first part of the thesis, this method is expected to work reliably as long as the network is dense enough such that the weighted average is taken over a sufficiently large amount of measurement data. As mentioned in Section 4.5, the interplay of a distributed network architecture, averaging of neighborhood information, and robustification make a thorough analysis of the effects of the network density an intricate task that is beyond the scope of this work.

10

Centralized Multi-Target Tracking

This chapter is concerned with multi-target tracking in a centralized sensor network with fusion center. To this end, a centralized version of the D-PPHDF is proposed in Section 10.1. Its computational complexity and communication load is investigated in Section 10.2.

10.1 The Multi-Sensor Particle PHD Filter

The proposed MS-PPHDF is a centralized, multi-sensor Particle PHD Filter that relies on a fusion center with access to the measurements of all nodes in the network. It is based on the formulation of the single-sensor Particle PHD Filter in [Vo et al., 2003; Clark, 2006; Hong et al., 2011] but with an extended measurement set comprising the measurements of the entire network. Hence, one might obtain more than one measurement per target—a change to the typical assumption in target tracking that each target produces *at most* one measurement [Bar-Shalom et al., 2011]. To account for this change, a pre-clustering step is added before the weighting step, and the weight update is changed accordingly. A similar partitioning of the measurement set is used in extended target tracking, where a sensor can receive multiple tar-

get reflections due to the target's physical extent [Granstrom & Orguner, 2012; Granstrom et al., 2014].

In the following, the individual steps of the algorithm are given in detail:

- Merging:
  The sets $\left\{\boldsymbol{s}^p_{\text{pers}}(t-1), w^p_{\text{pers}}(t-1)\right\}_{p=1}^{N_{\text{pers}}(t-1)}$ and $\{\boldsymbol{s}^p_{\text{new}}(t-1), w^p_{\text{new}}(t-1)\}_{p=1}^{N_{\text{new}}(t-1)}$ consist of the persistent and newborn particles, $\boldsymbol{s}^p_{\text{pers}}(t-1)$ and $\boldsymbol{s}^p_{\text{new}}(t-1)$, respectively, at time step $t-1$ with their respective weights $w^p_{\text{pers}}(t-1)$ and $w^p_{\text{new}}(t-1)$. These sets are merged to become the total set $\{\boldsymbol{s}^p_{\text{tot}}(t), w^p_{\text{tot}}(t)\}_{p=1}^{N_{\text{tot}}(t)}$ of particles and weights at time step t. Here, $N_{\text{tot}}(t)$ is the total number of particles at time step t, which is given by

  $$N_{\text{tot}}(t) = N_{\text{pers}}(t-1) + N_{\text{new}}(t-1),$$

  with $N_{\text{pers}}(t-1)$ and $N_{\text{new}}(t-1)$ denoting the respective number of persistent and newborn particles at the previous time step.

- Predicting:
  As in the D-PPHDF, each particle is propagated through the system model according to

  $$\boldsymbol{s}^p_{\text{pers}}(t) = \boldsymbol{F}\boldsymbol{s}^p_{\text{tot}}(t), \qquad p = 1, \ldots, N_{\text{pers}}(t) = N_{\text{tot}}(t), \tag{10.1}$$

  to become a persistent particle. The corresponding weights are multiplied with the probability of survival p_{S} as

  $$w^p_{\text{pers}}(t \mid t-1) = p_{\text{S}} w^p_{\text{tot}}(t), \qquad p = 1, \ldots, N_{\text{pers}}(t). \tag{10.2}$$

- Measuring:
  The sensor nodes obtain measurements of the targets.

- Pre-Clustering:
  Since there might be more than one measurement per target, the measurements of the entire network are pre-clustered before the weighting step and each measurement is assigned a label $C(\boldsymbol{z})$ that reflects the cardinality of its own cluster. This can be done, for instance, using single-linkage clustering [Everitt et al., 2011]. The clustering is based on the distance between measurements, i.e., spatially close measurements are assumed

to stem from the same target. Hence, when two or more targets are too close to each other, cardinality errors may occur.

- Weighting:
  All available target measurements, which comprise the set Σ_t, are used to update the persistent particle weights according to

$$w^p_{\text{pers}}(t) = \left[1 - p_{\text{D}} + \sum_{\boldsymbol{z}_j \in \Sigma_t} w^p_{j,\text{update}}(t)\right] w^p_{\text{pers}}(t \,|\, t-1), \tag{10.3}$$

  with

$$w^p_{j,\text{update}}(t) = \frac{p_{\text{D}} f_t(\boldsymbol{z}_j \,|\, \boldsymbol{x}^p(t))}{(\lambda_{\text{FA}} c_{\text{FA}}(\boldsymbol{z}_j) + \mathcal{L}(\boldsymbol{z}_j))\, C(\boldsymbol{z}_j)}, \tag{10.4}$$

  and

$$\mathcal{L}(\boldsymbol{z}_j) = \sum_{q=1}^{N_{\text{pers}}(t)} p_{\text{D}} f_t(\boldsymbol{z}_j \,|\, \boldsymbol{x}^q(t)) w^q_{\text{pers}}(t \,|\, t-1). \tag{10.5}$$

  Note that—in contrast to the D-PPHDF—the weighting step is applied only once using the entire set of measurements. Therefore—and since there might be more than one measurement per target—one has to ensure that the weight update terms $w^p_{j,\text{update}}$—and consequently the particle weights—still sum to the number of targets present. This is done by normalizing (10.4) with $C(\boldsymbol{z}_j)$—i.e., the cardinality of the cluster to which the current measurement $\boldsymbol{z}_j$ belongs.

  Afterwards, the set $\Sigma_{i,\text{cand}}$ of candidate measurements for the ATB step is formed according to

$$\Sigma_{i,\text{cand}} = \Sigma_t \backslash \left\{ \boldsymbol{z}_{m_p} \;\middle|\; m_p = \arg\max_j w^p_{j,\text{update}}(t), p = 1, \ldots, N_{\text{pers}}(t) \right\}. \tag{10.6}$$

- Resampling:
  The expected number of targets $\hat{N}_{\text{tgt}}(t)$ is calculated from the total persistent particle mass as

$$\hat{N}_{\text{tgt}}(t) = \left\lfloor \sum_{p=1}^{N_{\text{tot}}(t)} w^p_{\text{pers}}(t) \right\rfloor. \tag{10.7}$$

Consequently, the number of persistent particles is updated according to

$$N_{\text{pers}}(t) = \hat{N}_{\text{tgt}}(t) N_{\text{p}}.$$

Furthermore, the set of persistent particles is resampled by drawing $N_{\text{pers}}(t)$ particles with probability $\frac{w^p_{\text{pers}}(t)}{\hat{N}_{\text{tgt}}(t)}$. Then, the weights are reset to equal values as

$$w^p_{\text{pers}}(t) = \frac{\hat{N}_{\text{tgt}}(t)}{N_{\text{pers}}(t)}, \qquad p = 1, \ldots, N_{\text{pers}}(t). \tag{10.8}$$

- Clustering:
  In contrast to the D-PPHDF, there is only one estimate of the expected number of targets. Hence, *k-means clustering* [MacQueen, 1967] can be used to find the estimated target states by grouping the resampled particles into $\hat{N}_{\text{tgt}}(t)$ clusters and calculating the centroid of each cluster.

- Roughening:
  Roughening is performed analogously to the D-PPHDF.

- Adaptive Target Birth:
  N_{p} new particles are placed randomly around each candidate measurement $\boldsymbol{z}_j \in \Sigma_{i,\text{cand}}$ yielding a total number of $N_{\text{new}}(t) = N_{\text{p}} \cdot |\Sigma_{i,\text{cand}}|$ newborn particles. The corresponding weights are chosen according to

  $$w^p_{\text{new}}(t) = \frac{p_{\text{B}}}{N_{\text{new}}(t)}, \qquad p = 1, \ldots, N_{\text{new}}(t),$$

  where p_{B} is the probability of birth.

The pseudo-code of the MS-PPHDF is given in Appendix B.2.

10.2 Computational Complexity and Communication Load

In this section, the computational complexity and the communication load of the MS-PPHDF are analyzed. The following steps are performed at every time instant t but time dependency is omitted for simplicity:

- Prediction:
  The prediction step described by (10.1) and (10.2) is performed for each of the N_{tot} particles and is linear in the dimensionality d.
  $\Rightarrow \mathcal{O}(N_{\text{tot}} d)$

- Pre-Clustering:
  The pre-clustering step relies on single-linkage clustering. The complexity is therefore cubic in the total number of measurements N_{meas} [Murtagh, 1983].
  $\Rightarrow \mathcal{O}(N_{\text{meas}}^3)$

- Weighting:
  Each particle is updated in the weighting step given by (10.3)-(10.6). The weight update as well as the designation of candidate measurements for ATB depends on the number of measurements $N_{\text{meas}} = |\Sigma|$.
  $\Rightarrow \mathcal{O}(N_{\text{tot}} N_{\text{meas}})$

- Resampling:
  The estimation of the number of targets and the resampling step in (10.7)-(10.8) are linear in the number of particles used for the calculation [Gustafsson, 2010].
  $\Rightarrow \mathcal{O}(N_{\text{tot}} + N_{\text{tgt}} N_{\text{p}})$

- Clustering:
  In contrast to the D-PPHDF, k-means clustering is used. The complexity of Lloyd's implementation is given in [Inaba et al., 1994] as
  $\Rightarrow \mathcal{O}((N_{\text{tgt}} N_{\text{p}})^{d N_{\text{tgt}}+1} \log(N_{\text{tgt}} N_{\text{p}}))$

- Roughening:
  Roughening is linear in the dimensionality of the particles and their number.
  $\Rightarrow \mathcal{O}(N_{\text{tgt}} N_{\text{p}} d)$

- Adaptive Target Birth:
  The birth process depends on the number of particles per target N_{p} as well as the number of candidate measurements N_{cand}.
  $\Rightarrow \mathcal{O}(N_{\text{p}} N_{\text{cand}})$

To summarize, the computational complexity of the MS-PPHDF is largely comparable to that of the D-PPHDF. The only exception is the pre-clustering step, which scales cubicly with the total number of measurements and adds additional complexity to the algorithm. As

a tradeoff, the communication load of the MS-PPHDF clearly is lower compared to the D-PPHDF because there is only the initial transmission of measurements from the nodes to the fusion center. However—considering a setup with relatively small communication radii—, this initial communication step requires a lot of relaying and leads to high traffic density in the vicinity of the fusion center. Furthermore, this communication structure exhibits a single point of failure while a distributed sensor network is inherently redundant.

11

Performance Evaluation

This chapter is dedicated to the performance evaluation of the proposed distributed and centralized tracking algorithms. The trackers are compared to the alternative distributed Particle PHD Filter from [Uney et al., 2010], which will be referred to as Distributed Data Fusion Particle PHD Filter (DDF-PPHDF). Here, each node runs its own Particle PHD Filter using only its own measurements. In a subsequent step, the particles are distributed over the neighborhood and reweighted by fusing their corresponding Exponential Mixture Densities.

In Section 11.1, a review of the Optimal Subpattern Assignment (OSPA) metric is given, which will serve as a performance metric in the simulations. Furthermore, a distributed version of the Posterior Cramér-Rao Lower Bound (PCRLB) is derived in Section 11.2 to benchmark the performance of the different tracking algorithms. Section 11.3 presents the simulation results under Gaussian measurement noise of different variance. Furthermore, ε-contaminated noise with different contamination ratios is considered to examine the robustness of the algorithms. In addition, the performance for different amounts of clutter is analyzed.

11.1 The Optimal Subpattern Assignment Metric

The Optimal Subpattern Assignment (OSPA) metric is a consistent metric for the performance-evaluation of multi-object filters [Schuhmacher et al., 2008]. Assume that Π_k is the set of all permutations on $\{1, 2, ..., k\}$ for $k \in \mathbb{N}$, and $X = \{x_1, ..., x_m\}$ as well as $Y = \{y_1, ..., y_n\}$ are finite subsets of the closed and bounded observation window $W \subset \mathbb{R}^N$ with $m, n \in \mathbb{N}_0, m \leq n$. The pth-order OSPA metric $\bar{d}_p^{(c)}$ with cut-off $c > 0$ and $1 \leq p < \infty$ is calculated as

$$\bar{d}_p^{(c)}(X, Y) = \left(\frac{1}{n} \left(\min_{\pi \in \Pi_k} \sum_{i=1}^{m} d^{(c)}(x_i, y_{\pi(i)})^p + c^p(n - m) \right) \right)^{\frac{1}{p}},$$

where

$$d^{(c)}(x, y) = \min(c, d(x, y)),$$

is an arbitrary distance metric—e.g., the Euclidean distance—between $x, y \in W$ with an upper bound at the value c. In other words, the OSPA metric finds the m-point subpattern or subset of Y that is closest to X in terms of the pth-order Wasserstein distance [Hoffmann & Mahler, 2004], resulting in an optimal point assignment. Each point is assigned the corresponding distance value cut off at c. Points in Y without a corresponding partner in X are simply assigned the cut-off value c. Finally, the pth-order average of all values is computed.

11.2 The Distributed Posterior Cramér-Rao Lower Bound

Rather than evaluating the performance of the different multi-target tracking algorithms solely based on an error metric, it makes more sense to derive a minimum variance bound on the estimation error, which enables an absolute performance evaluation. For time-invariant statistical models, the most commonly used bound is the Cramér-Rao Lower Bound, which is given by the inverse of Fisher's information matrix [Kay, 1993]. In [Braca et al., 2012] and [Braca et al., 2013], the Cramér-Rao Lower Bound is used in the context of multi-sensor multi-target

tracking of an unknown number of unlabeled targets in order to evaluate the performance as well as prove the asymptotic efficiency of the PHD as the number of nodes goes to infinity. Since the focus of this work is on the tracking behavior of a fixed network over time, the Posterior Cramér-Rao Lower Bound (PCRLB)—an extension of the Cramér-Rao Lower Bound for the time-variant case [Van Trees, 2004]—is considered in the sequel. This bound can be calculated sequentially with the help of a Riccati-like recursion derived in [Tichavsky et al., 1998]. Furthermore, in [Hue et al., 2002a; Hue et al., 2002b; Hue et al., 2002c], the PCRLB is adapted for a multi-target tracking scenario in which the tracker can obtain more than one measurement per target.

Let $\pi_t^m, m = 1, \ldots, M$ denote the probability that any measurement is associated with target m at time instant t. With the corresponding stochastic process Π_t^m, the new stochastic process of association probabilities and target states to be estimated becomes $\mathbf{\Phi}_t = \left(\Pi_t^{1:M}, \mathbf{\Xi}_t^{1:M}\right)$. Fisher's information matrix $J_{\mathbf{\Phi}_t} = \begin{bmatrix} J_{\Pi_t} & J_{\Pi_t}^{\Xi_t} \\ J_{\Xi_t}^{\Pi_t} & J_{\Xi_t} \end{bmatrix}$ can now be formed as described in [Hue et al., 2002a; Hue et al., 2002b; Hue et al., 2002c]. However, as the number of targets varies over time—i.e., targets might enter or exit the region of interest—, $J_{\mathbf{\Phi}_t}$ has to be expanded or shrunk in the inverse matrix domain as described in [Bessell et al., 2003]. The PCRLB B_t at time instant t can be obtained as the trace of the inverted submatrix J_{Ξ_t} according to

$$B_t = \text{trace}\left\{\left[J_{\Xi_t} - J_{\Xi_t}^{\Pi_t} J_{\Pi_t}^{-1} J_{\Pi_t}^{\Xi_t}\right]^{-1}\right\}.$$

Note that—in a distributed multi-target tracking scenario—B_t corresponds to a lower bound on the estimation error of a central processing unit with access to all measurements. To account for completely distributed multi-target tracking scenarios with in-network processing, the PCRLB is extended to the Distributed Posterior Cramér-Rao Lower Bound (DPCRLB) as follows.

To obtain the DPCRLB, each node k computes its own PCRLB B_t^k considering only the measurements of its two-hop neighborhood, which is given by

$$\mathcal{N}_k^{(2)} = \bigcup_{l \in \mathcal{N}_k} \mathcal{N}_l,$$

i.e., the neighbors of node k and their neighbors. Furthermore, only the targets within the

Variable	Value	Description
Δt	1 s	time step of the tracking algorithm
N	30	number of nodes
N_{MC}	1,000	number of Monte Carlo runs
σ_r^2	$0.1, 0.3\,\text{m}^2$	componentwise power of measurement noise
σ_q^2	$0.01\,\text{m}^2$	componentwise power of state noise
ε	$0.1, 0.3$	contamination ratio
r_{com}	$2r_{\text{sen}}$	communication radius
r_{sen}	6 m	sensing radius
E_c	6	empirical interval length for jitter
K	0.2	tuning constant for roughening
N_{p}	500	number of particles per target
p_{B}	0.8	probability of birth
p_{D}	0.95	probability of detection
p_{S}	0.98	probability of survival
λ_{FA}	$0.1, 0.3$	average number of false alarms / clutter
$c_{\text{FA}}(\boldsymbol{z})$	$\frac{1}{\pi r_{\text{sen}}^2}$	PDF of false alarms / clutter (uniform)
c	2	cut-off value of the OSPA metric
p	2	order of the OSPA metric

Table II.1: Overview of the parameters used in the simulations for evaluating the performance of the proposed tracking algorithms.

sensing range of $\mathcal{N}_k$ are taken into account. Clearly, only nodes with a neighborhood in the vicinity of at least one target will be able to calculate a PCRLB. The DPCRLB $B_{t,\text{dist}}$ at time instant t is then obtained by averaging over these values according to

$$B_{t,\text{dist}} = \frac{1}{|\mathcal{M}|} \sum_{k \in \mathcal{M}} B_t^k,$$

where $\mathcal{M}$ is the set of all nodes that are able to compute a PCRLB.

11.3 Simulations

In the following simulations, a static sensor network as depicted in Figure 9.1 is used to perform multi-target tracking. The network is centered around the point of origin $[0, 0]^\top$ and distributed such that 1-coverage of the region of interest is guaranteed. It covers an area of ap-

proximately $2500\ \mathrm{m}^2$. Clutter is assumed Poisson and uniformly distributed over the sensing range of each node with an average rate of $\lambda_{\mathrm{FA}} = 0.1$ and 0.3. Moreover, Gaussian measurement noise with variance $\sigma_r^2 = 0.1$ and 0.3 as well as ε-contaminated noise with a ten-times higher variance and a contamination rate of $\varepsilon = 0.1$ and 0.3 are considered. The objective is to track three targets for $t = 0, \ldots, 30$. The targets enter the region of interest at time steps $t \in \{0, 9, 14\}$ from the north, south, and west, respectively. The target trajectories as shown in Figure 9.1 are deterministic as is often the case in target tracking simulations. This guarantees the comparability of the different Monte Carlo runs regarding, for instance, the number of targets present [Bessell et al., 2003]. $N_{\mathrm{MC}} = 1{,}000$ Monte Carlo runs are performed to evaluate the performance of the tracking algorithms in terms of the OSPA metric and the estimated number of targets. The latter is computed with respect to the joint set of target state estimates of the entire active network, which is found by clustering the target state estimates of all active nodes. To be able to use the DPCRLB as a benchmark, the OSPA metric is squared and scaled by the number of targets as proposed in [Braca et al., 2013]. For the sake of simplicity, collisions between targets and sensor nodes are neglected. An overview of all simulation parameters is given in Table 11.1.

In the following, three different simulations are performed. In Section 11.3.1, the performance of the MS-PPHDF and the D-PPHDF is compared to that of the alternative DDF-PPHDF and to the DPCRLB. Measurement noise is zero-mean Gaussian with variance $\sigma_r^2 = 0.1, 0.3$ and the clutter rate is fixed to 0.1. The second simulation in Section 11.3.2 evaluates the performance of the MS-PPHDF, the D-PPHDF, and the DDF-PPHDF under a higher clutter rate of 0.3. The final simulation in Section 11.3.3 is dedicated to examining the robustness of the tracking algorithms in the face of ε-contaminated noise with 10% and 30% contamination and different clutter rates. To this end, the D-PPHDF is not only compared to the MS-PPHDF and the DDF-PPHDF but also to its robust variants Median-D-PPHDF and M-D-PPHDF.

11.3.1 Simulation I: Gaussian Noise and Low Clutter

The simulation results are depicted in Figure 11.1. While the upper row considers zero-mean Gaussian measurement noise with a per-component variance of $\sigma_r^2 = 0.1$, the lower row shows the results for $\sigma_r^2 = 0.3$. The left column shows the squared and scaled OSPA metric

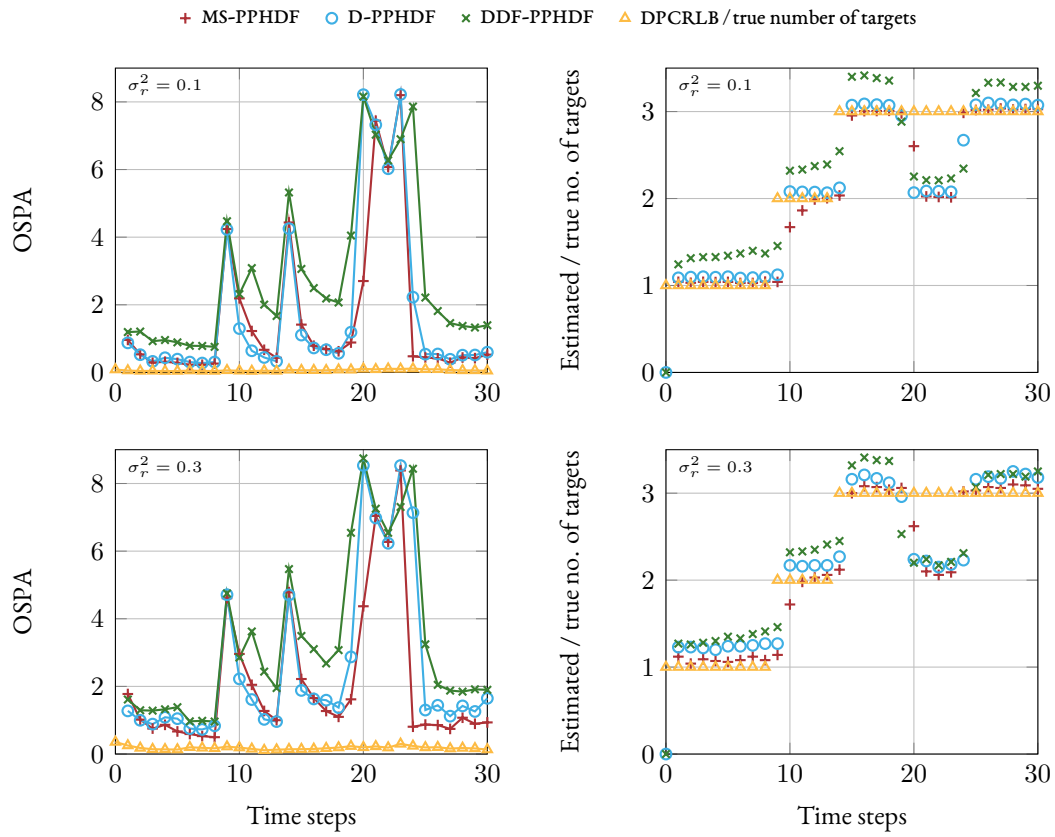

Figure II.1: Results for Simulation I with Gaussian noise of variance $\sigma_r^2 = 0.1, 0.3$ and clutter rate $\lambda_{\text{FA}} = 0.1$. The left part of the figure shows the squared and scaled OSPA metric for each algorithm compared to the DPCRLB, while the right part compares the estimated to the true number of targets.

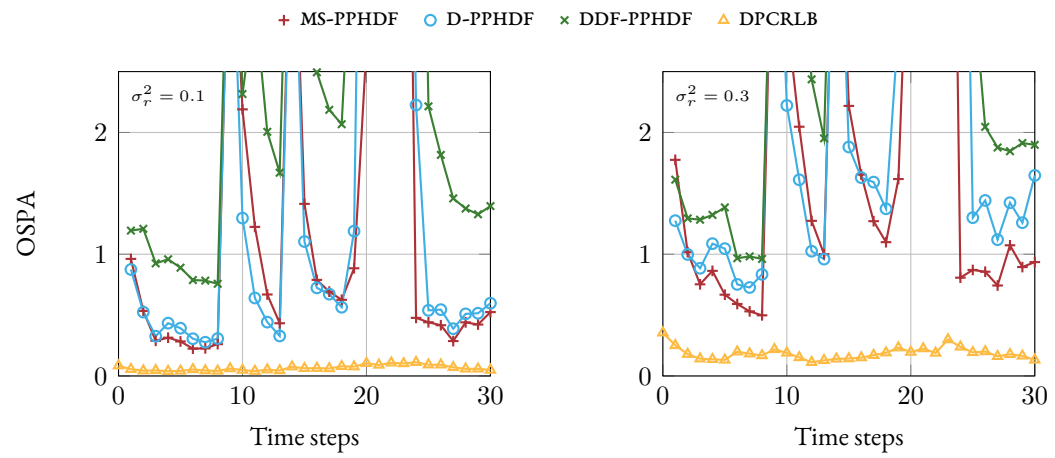

Figure II.2: Results for Simulation I with Gaussian noise of variance $\sigma_r^2 = 0.1, 0.3$ and clutter rate $\lambda_{\text{FA}} = 0.1$ (zoomed in). The squared and scaled OSPA metric of each algorithm is compared to the DPCRLB.

over time compared to the DPCRLB. The estimated number of targets is depicted in the right column. Since the OSPA metric contains a penalty for an erroneous estimate of the number of targets, this side-by-side comparison facilitates the interpretation of the tracking results.

First, consider the case of $\sigma_r^2 = 0.1$ in the upper row. It can be observed that neither tracking algorithm provides an OSPA value or an estimate of the number of targets for $t = 0$. This is expected and due to ATB, which initializes new particle clouds based on the measurements from the previous time step. Thus, target birth is delayed by one time step and tracking can only be performed for $t > 0$. The same effect can be witnessed at $t = 9$ and $t = 14$, respectively, which mark the time instants at which targets 2 and 3 enter the region of interest. Here, the OSPA curves of all trackers exhibit a spike, which is due to the fact that the newborn particles are not yet considered in the tracker and, hence, the number of estimated targets is too low.

Another sudden rise of all the OSPA curves can be observed in the time interval $20 \leq t \leq 24$ with a valley at $t = 22$. Looking at the estimated number of targets, this phenomenon can be attributed to the fact that only two of the three targets are recognized by the tracking algorithms. Since the target trajectories are deterministic, targets 2 and 3 are known to cross paths in the given time interval. Due to the inability of the clustering algorithm to separate strongly overlapping sets of measurements, the two targets merge into one as long as they are close to each other. When the two targets occupy almost exactly the same position—i.e., at $i = 22$—, the OSPA metric decreases due to the decrease in measurement variance. As the targets drift apart, the variance and with it the OSPA metric increases up to the point where the two targets can be recognized as separate again and the corresponding penalty is switched off.

From the left picture in the upper row it is evident that the centralized MS-PPHDF and the distributed D-PPHDF achieve approximately the same performance with OSPA values closely approaching the DPCRLB when the number of targets stays constant. Moreover, both algorithms deliver very accurate estimates of the number of targets—given they are separable by clustering—as can be seen from the right picture. The DDF-PPHDF, however, continuously exhibits a worse performance than the D-PPHDF, both in terms of the OSPA metric as well as the estimated number of targets. This is where the additional communication in the proposed D-PPHDF shows its strength in reducing uncertainty due to measurement noise

and clutter. Apart from achieving worse tracking results, the DDF-PPHDF also has more difficulty in separating targets 1 and 2 when they cross paths, resulting in an earlier rise and a later fall of the OSPA metric, compared to the proposed approach.

In the case of $\sigma_r^2 = 0.3$—as visible in the lower row—the overall performance of the different tracking algorithms is very similar to the case of $\sigma_r^2 = 0.1$. Figure 11.2 provides zoomed-in versions of the OSPA results from Figure 11.1, which simplify the comparison of the different algorithms by neglecting the impact of the penalty due to an erroneous estimate of the number of targets. One can observe that the value of the DPCRLB is always smaller or equal to the respective measurement variance. As stated before, the centralized MS-PPHDF and the distributed D-PPHDF exhibit very similar performance and deliver better tracking results than the DDF-PPHDF. While the MS-PPHDF achieves lower OSPA values than the D-PPHDF when the number of targets stays constant—i.e., for $3 \leq t \leq 8$ and $24 \leq t \leq 30$—, the D-PPHDF performs better directly after a new target appears—i.e., for $1 \leq t \leq 2$, $10 \leq t \leq 13$, and $15 \leq t \leq 18$. This is likely due to the fact that the two-step communication scheme employed in the D-PPHDF is able to reduce the impact of measurement noise and clutter faster than the centralized MS-PPHDF can.

In the righthand picture it can be observed that the higher measurement noise of variance $\sigma_r^2 = 0.3$ affects the performance of all algorithms, resulting in higher OSPA curves. While the OSPA curves of the MS-PPHDF and the DDF-PPHDF are proportionally shifted upward by approximately the same value—i.e., they are equally impacted by the higher noise level—, the D-PPHDF seems to be slightly more affected by the change. However it still outperforms the DDF-PPHDF at all time instants.

To summarize, the proposed D-PPHDF yields better performance than the existing DDF-PPHDF in estimating the number of targets and tracking them, irrespective of the amount of measurement noise. In addition, it is also a bit faster in delivering correct state estimates of new targets than the centralized MS-PPHDF and performs only slightly worse once the number of targets stays constant.

11.3.2 Simulation II: Gaussian Noise and Higher Clutter

The simulation results are shown in Figure 11.3. The upper row considers zero-mean Gaussian measurement noise with a per-component variance of $\sigma_r^2 = 0.1$ while the lower row shows

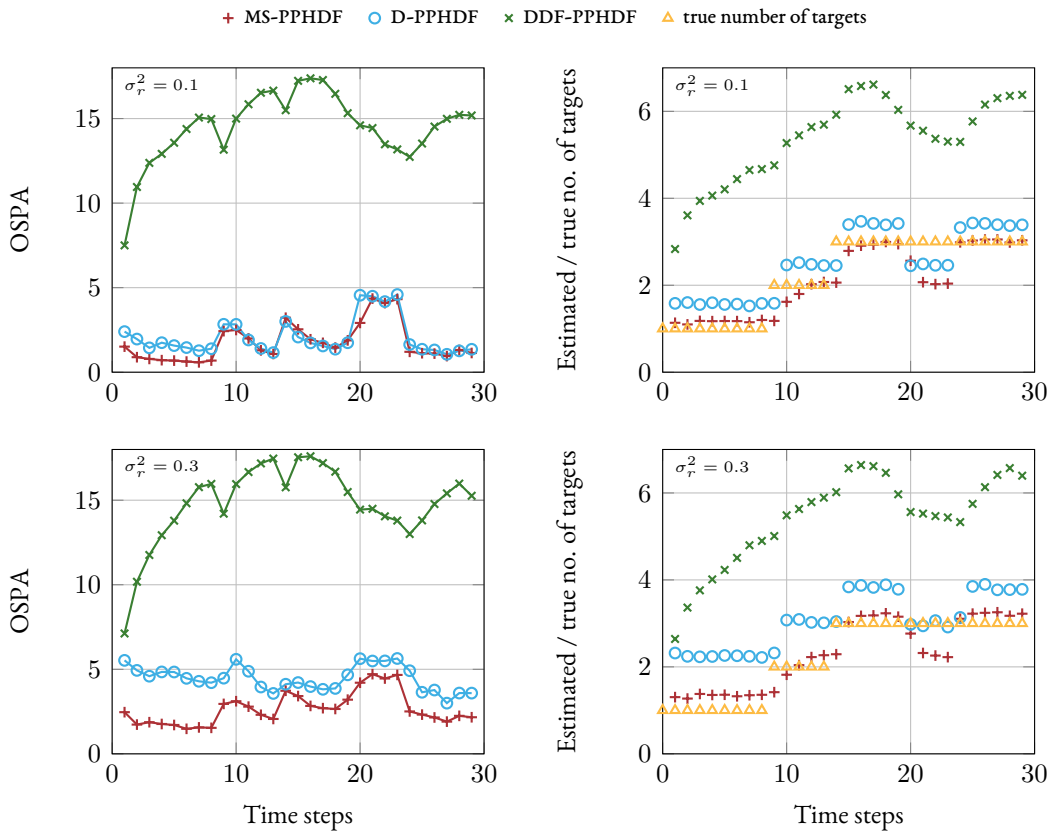

Figure 11.3: Results for Simulation II with Gaussian noise and clutter rate $\lambda_{\text{FA}} = 0.3$. The left part of the figure shows the squared and scaled OSPA metric for each algorithm compared to the DPCRLB while the right part compares the estimated to the true number of targets.

the results for $\sigma_r^2 = 0.3$.

Although the higher clutter rate causes an increase in the OSPA value of all algorithms, the MS-PPHDF is still able to correctly estimate the number of targets—except for the crossing period $20 \leq t \leq 24$—in both cases. When rounding the estimate to the next lower integer, the D-PPHDF also yields acceptable results for $\sigma_r^2 = 0.1$. For $\sigma_r^2 = 0.3$ the number of targets is overestimated by one for $1 \leq t \leq 15$, causing a stronger degradation of the scaled and squared OSPA value in this interval.

Apparently, the DDF-PPHDF is not able to cope with a clutter rate of 0.3 as the number of targets is largely overestimated. Hence, no accurate target tracking is possible.

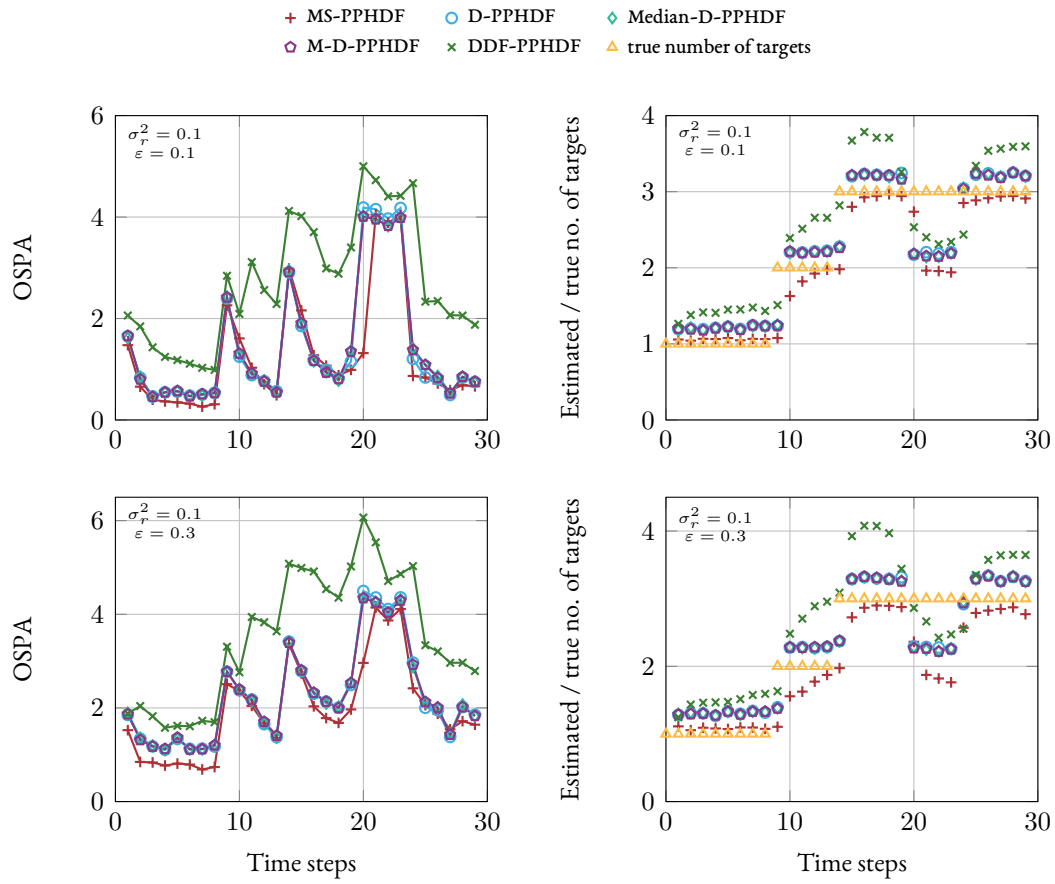

Figure 11.4: Results for Simulation III with ε-contaminated noise, measurement variance $\sigma_r^2 = 0.1$, and clutter rate $\lambda_{\text{FA}} = 0.1$. The upper row considers $\varepsilon = 0.1$, the lower row shows the results for $\varepsilon = 0.3$.

11.3.3 Simulation III: Robustness Evaluation

In the third simulation, the performance of the MS-PPHDF, the D-PPHDF, and the DDF-PPHDF are compared in the face of ε-contaminated noise with different amounts of contamination. In addition, the robustified versions of the D-PPHDF—the Median-D-PPHDF and the M-D-PPHDF—are considered. The simulation results for clutter rates $\lambda_{\text{FA}} = 0.1$ are given in Figure 11.4 and Figure 11.5, with $\sigma_r^2 = 0.1$ and $\sigma_r^2 = 0.3$, respectively. The results for $\lambda_{\text{FA}} = 0.3$ are depicted in Figure 11.6 and Figure 11.7. The top row of each figure considers a noise contamination of 10 %, the lower row shows the results for 30 %.

Consider the case of $\lambda_{\text{FA}} = 0.1$ first. It can be observed that the centralized MS-PPHDF is still the best performing algorithm, being largely unaffected by higher noise variance and

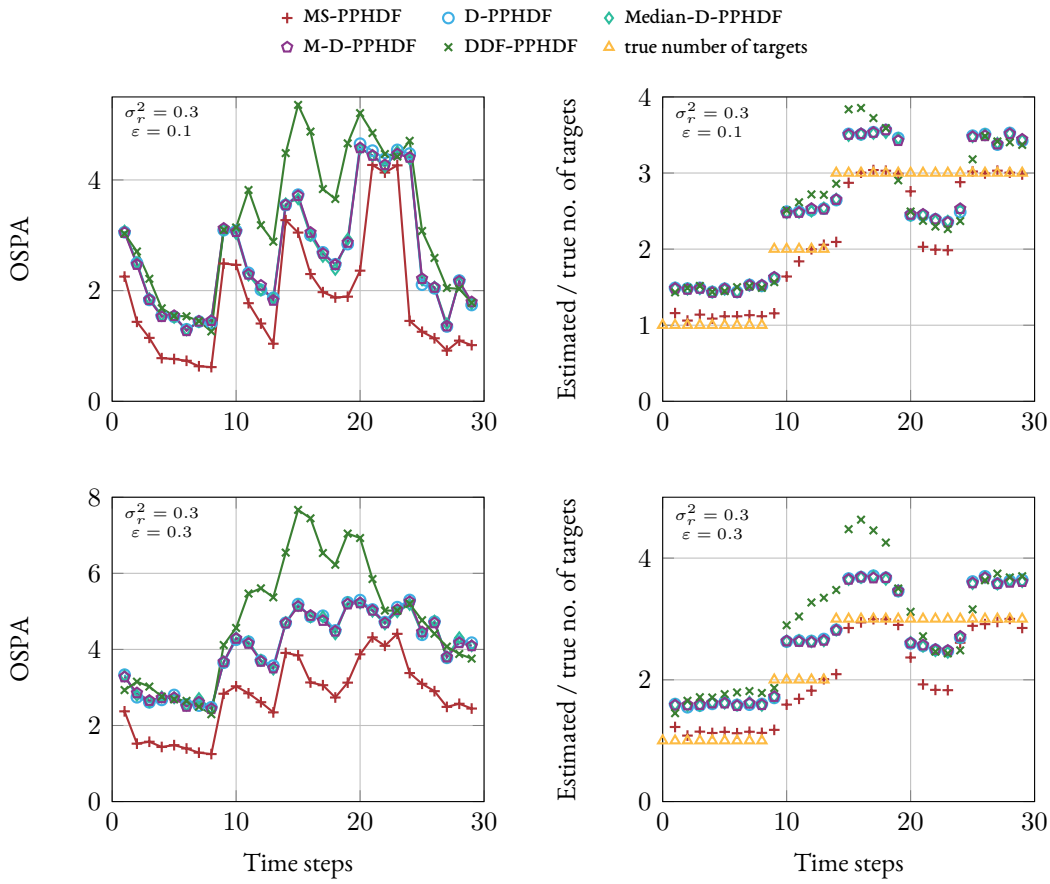

Figure 11.5: Results for Simulation III with ε-contaminated noise, measurement variance $\sigma_r^2 = 0.3$, and clutter rate $\lambda_{\text{FA}} = 0.1$. The upper row considers $\varepsilon = 0.1$, the lower row shows the results for $\varepsilon = 0.3$.

outliers. The D-PPHDF and its robust variants are a close second, being primarily affected by the higher clutter rate and the higher noise variance. They only exhibit a slight additional performance degradation when the noise contamination increases to 30 %. The performance of the Median-D-PPHDF and the M-D-PPHDF is equally good, with a marginal improvement over the standard D-PPHDF at the crossing of targets two and three in the time interval $20 \leq t \leq 24$. Hence, it can be said that the MS-PPHDF and the D-PPHDF are robust by design and can handle a fraction of at least 10 % in the given scenario. As the performance is already satisfactory, using robust estimators in the D-PPHDF only provides a marginal improvement.

The DDF-PPHDF, in contrast, is more severely affected by outliers. Both the OSPA value and the estimated number of targets increase with the introduction of noise contamination.

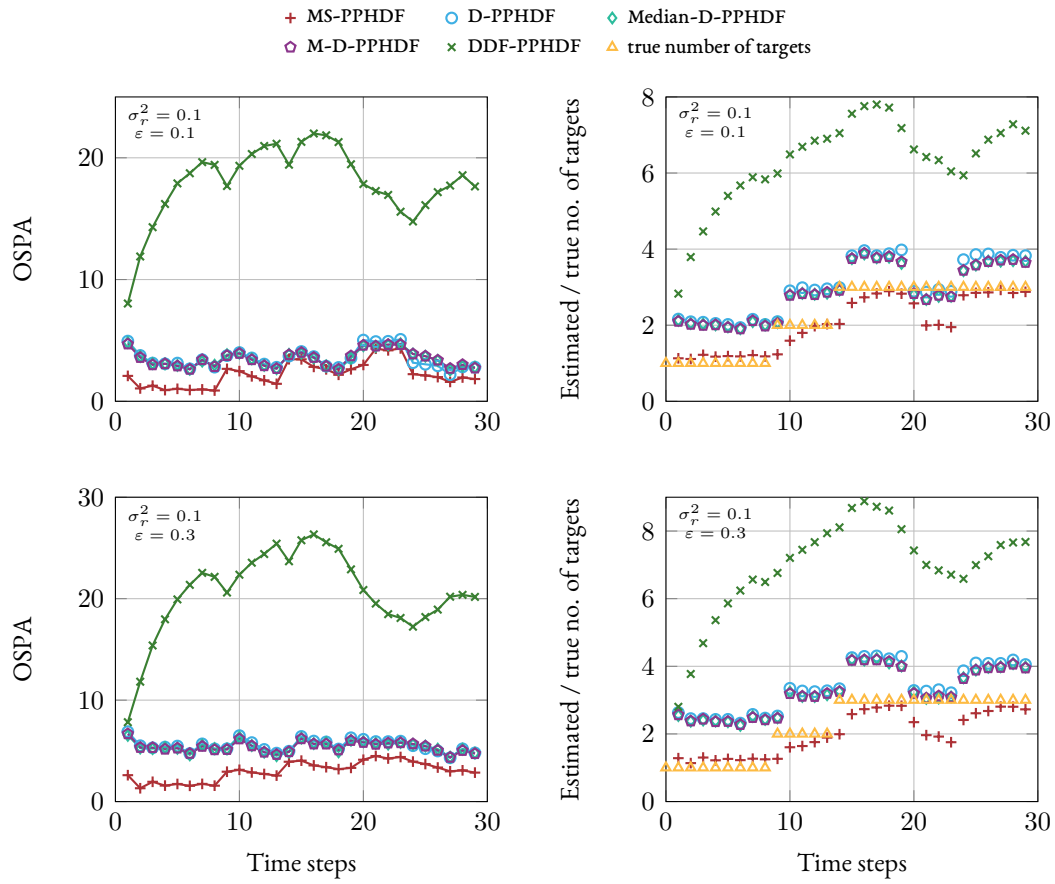

Figure II.6: Results for Simulation III with ε-contaminated noise, measurement variance $\sigma_r^2 = 0.1$, and clutter rate $\lambda_{\text{FA}} = 0.3$. The upper row considers $\varepsilon = 0.1$, the lower row shows the results for $\varepsilon = 0.3$.

When the number of targets remains constant and no target crossing takes place—i.e., for $t < 9$ and $t > 24$—, the number of targets is only slightly overestimated. However, when targets two and three enter the scene—i.e., for $10 \leq t < 20$—, the estimate is inaccurate, which imposes a penalty on the scaled and squared OSPA metric. Hence, the DDF-PPHDF is not a robust algorithm for the considered tracking scenario.

In the case of $\lambda_{\text{FA}} = 0.3$, the DDF-PPHDF, again, breaks down completely. The MS-PPHDF, however, is still able to give accurate results with only slight deviations from the true number of targets and a small OSPA value. Unfortunately, the combination of noise contamination and more clutter is too much for the D-PPHDF and its variants to handle. They overestimate the number of targets by one to two, causing the OSPA value to rise as well.

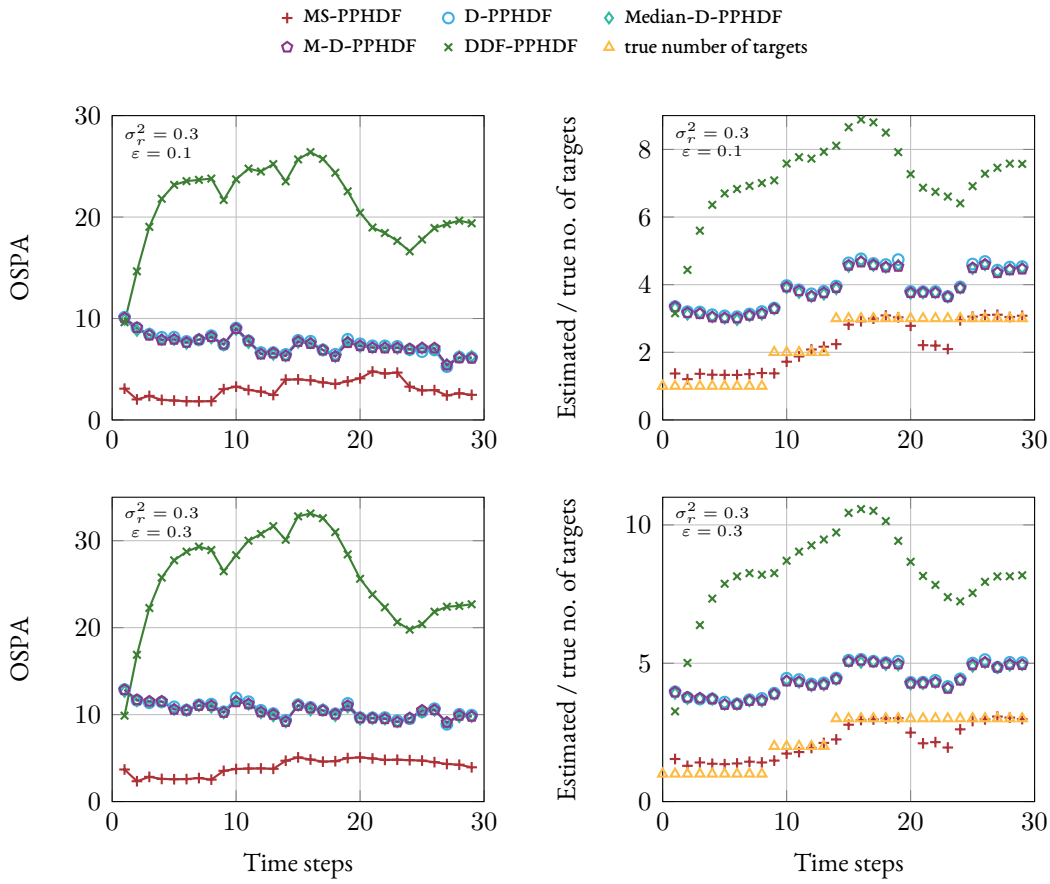

Figure 11.7: Results for Simulation III with ε-contaminated noise, measurement variance $\sigma_r^2 = 0.3$, and clutter rate $\lambda_{\text{FA}} = 0.3$. The upper row considers $\varepsilon = 0.1$, the lower row shows the results for $\varepsilon = 0.3$.

To summarize, the proposed MS-PPHDF and D-PPHDF are—to a certain extent—robust against outliers of the ε-contamination kind. This property is due to the employed two-way communication scheme, which vets measurements as well as intermediate target location estimates against the entire network or the neighborhood of each node. Therefore, adding additional robustness via robust estimators, which also leverage the power of neighborhood communication, does not deliver a noticeable performance improvement. The DDF-PPHDF is not robust and breaks down in the face of outliers.

11.4 Summary

In the second part of this thesis, a distributed and a centralized Particle PHD Filter for multi-target tracking in sensor networks were developed. Furthermore, a method for robustifying the distributed D-PPHDF by means of robust estimators was presented and two robust variants of the algorithm were introduced. In order to evaluate the performance of the proposed algorithms, a distributed version of the PCRLB was derived as a benchmark. Simulations revealed that the distributed D-PPHDF is faster in correctly tracking new targets than the centralized MS-PPHDF and performs only slightly worse when the number of targets stays constant. In addition, it delivers accurate tracking results as long as the targets are far enough apart so that their corresponding measurement clouds are separable. The proposed distributed tracker outperforms the existing DDF-PPHDF at the cost of additional communication between sensor nodes. Moreover, both the MS-PPHDF and the D-PPHDF are inherently robust against outliers; adding additional robustification is, hence, not necessary. In addition, the centralized MS-PPHDF is even able to handle higher clutter rates. The existing DDF-PPHDF, in contrast, is neither robust nor able to cope with more than 10 % clutter.

12

Conclusions & Outlook

The second part of this dissertation investigated the problem of multi-target tracking in a distributed sensor network. A distributed Particle PHD Filter for tracking multiple targets was developed and an extension for centralized network architectures was given. The proposed algorithms were shown to be robust against outliers by design, rendering the presented robustification steps optional. Furthermore, they continuously outperformed the only existing alternative, which—in addition—is non-robust and sensitive to clutter. Moreover, a novel distributed performance bound was derived for assessing the quality of the obtained location estimates.

The following problems constitute possible extensions of the presented work.

12.1 More Sophisticated Tracking Scenarios

When a new algorithm is developed, it is common to verify its functionality first in a simplified test case. The reasoning behind this is the elimination of any outside factors that could impact the algorithm's performance and lead to wrong conclusions. Therefore, the simulations in Chapter 11 were performed with a high probability of detection, relatively low clutter

rates, and a linear observation model as is common in the tracking literature (see, e.g. [Vo et al., 2005; Vo & Ma, 2006; Clark, 2006]). Future work should study more sophisticated scenarios to define possible breakdown points of the proposed algorithms. Furthermore, the consideration of mobile networks would enable reactions such as target pursuit or escape.

12.2 A Multi-Sensor Probability Hypothesis Density

The underlying principle of the proposed distributed tracking algorithms is the extension of the single-sensor PHD Filter to a network with multiple sensors. This is in line with existing approaches found in the literature—e.g., in [Uney et al., 2010; Uney et al., 2013; Battistelli et al., 2013]—although there are key differences in the considered network architecture and communication scheme. A more rigorous approach for multi-target tracking with multiple sensors is to consider a multi-sensor PHD Filter [Mahler, 2009; Mahler, 2010], which seeks to estimate and track a single *multi-sensor* PHD instead of multiple *single-sensor* PHDs. It would be interesting to see if the proposed algorithms can be adapted to this concept and examine the implications on track quality as well as communication load.

PART III

Appendix

A

Appendix to Part I

A.1 Mean and Variance of the Log-likelihood Ratio

The mean and the variance of the log-likelihood ratio are derived under the null hypothesis. The derivation under the alternative is analogous. To this end, the following identities are used:

$$\mathbb{E}[y^2] = \mu_y^2 + \sigma_y^2, \qquad \mathbb{E}[y^3] = \mu_y^3 + 3\mu_y\sigma_y^2, \qquad \mathbb{E}[y^4] = \mu_y^4 + 6\mu_y^2\sigma_y^2 + 3\sigma_y^4.$$

Note that the superscript k is dropped since the measurements at each agent are assumed to be independent and identically distributed. For the sake of simplicity, the time dependence is omitted as well.

$$\begin{aligned}
\mu_{\eta,0} &= \mathbb{E}_0\left[\frac{\sigma_1^2\left(y-\mu_0\right)^2 - \sigma_0^2\left(y-\mu_1\right)^2}{2\sigma_0^2\sigma_1^2} + \log\left(\frac{\sigma_0}{\sigma_1}\right)\right], \\
&= \mathbb{E}_0\left[\frac{\sigma_1^2\left(y^2 - 2\mu_0 y + \mu_0^2\right) - \sigma_0^2\left(y^2 - 2\mu_1 y + \mu_1^2\right)}{2\sigma_0^2\sigma_1^2} + \log\left(\frac{\sigma_0}{\sigma_1}\right)\right].
\end{aligned}$$

$$
\begin{aligned}
\mu_{\eta,0} &= \frac{\sigma_1^2\left(\mathbb{E}_0[y^2]-2\mu_0\mathbb{E}_0[y]+\mu_0^2\right)}{2\sigma_0^2\sigma_1^2} - \frac{\sigma_0^2\left(\mathbb{E}_0[y^2]-2\mu_1\mathbb{E}_0[y]+\mu_1^2\right)}{2\sigma_0^2\sigma_1^2} + \log\left(\frac{\sigma_0}{\sigma_1}\right), \\
&= \frac{\sigma_1^2\left(\mu_0^2+\sigma_0^2-2\mu_0^2+\mu_0^2\right)}{2\sigma_0^2\sigma_1^2} - \frac{\sigma_0^2\left(\mu_0^2+\sigma_0^2-2\mu_0\mu_1+\mu_1^2\right)}{2\sigma_0^2\sigma_1^2} + \log\left(\frac{\sigma_0}{\sigma_1}\right), \\
&= \frac{\sigma_1^2\sigma_0^2-\sigma_0^2\left(\mu_0^2+\sigma_0^2-2\mu_0\mu_1+\mu_1^2\right)}{2\sigma_0^2\sigma_1^2} + \log\left(\frac{\sigma_0}{\sigma_1}\right), \\
&= \underbrace{-\frac{\mu_0^2+\mu_1^2-2\mu_0\mu_1+\sigma_0^2-\sigma_1^2}{2\sigma_1^2}}_{Z_1} + \underbrace{\log\left(\frac{\sigma_0}{\sigma_1}\right)}_{Z_2}.
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{E}_0[\eta^2] &= \mathbb{E}_0\left[\left(\frac{\sigma_1^2\left(y-\mu_0\right)^2-\sigma_0^2\left(y-\mu_1\right)^2}{2\sigma_0^2\sigma_1^2} + \log\left(\frac{\sigma_0}{\sigma_1}\right)\right)^2\right], \\
&= \mathbb{E}_0\left[\frac{\left(\sigma_1^2\left(y-\mu_0\right)^2-\sigma_0^2\left(y-\mu_1\right)^2\right)^2}{4\sigma_0^4\sigma_1^4} + \log\left(\frac{\sigma_0}{\sigma_1}\right)^2\right. \\
&\qquad \left. + 2\,\frac{\sigma_1^2\left(y-\mu_0\right)^2-\sigma_0^2\left(y-\mu_1\right)^2}{2\sigma_0^2\sigma_1^2}\log\left(\frac{\sigma_0}{\sigma_1}\right)\right], \\
&= \overbrace{\mathbb{E}_0\left[\frac{\left(\sigma_1^2\left(y-\mu_0\right)^2-\sigma_0^2\left(y-\mu_1\right)^2\right)^2}{4\sigma_0^4\sigma_1^4}\right]}^{Z_3} + \log\left(\frac{\sigma_0}{\sigma_1}\right)^2 \\
&\qquad + 2\,\mathbb{E}_0\left[\frac{\sigma_1^2\left(y-\mu_0\right)^2-\sigma_0^2\left(y-\mu_1\right)^2}{2\sigma_0^2\sigma_1^2}\right]\log\left(\frac{\sigma_0}{\sigma_1}\right), \\
&= Z_3 + Z_2^2 + 2Z_1Z_2.
\end{aligned}
$$

$$
\begin{aligned}
Z_3 &= \mathbb{E}_0\left[\frac{\left(\sigma_1^2\left(y-\mu_0\right)^2-\sigma_0^2\left(y-\mu_1\right)^2\right)^2}{4\sigma_0^4\sigma_1^4}\right], \\
&= \mathbb{E}_0\left[\frac{\left(\sigma_1^2\left(y^2-2\mu_0y+\mu_0^2\right)-\sigma_0^2\left(y^2-2\mu_1y+\mu_1^2\right)\right)^2}{4\sigma_0^4\sigma_1^4}\right].
\end{aligned}
$$

$$
\begin{aligned}
Z_3 &= \mathbb{E}_0\left[\frac{\sigma_1^4\left(y^2-2\mu_0 y+\mu_0^2\right)^2+\sigma_0^4\left(y^2-2\mu_1 y+\mu_1^2\right)^2}{4\sigma_0^4\sigma_1^4}\right] \\
&\quad\times \mathbb{E}_0\left[\frac{\left(y^2-2\mu_0 y+\mu_0^2\right)\left(y^2-2\mu_1 y+\mu_1^2\right)}{2\sigma_0^2\sigma_1^2}\right], \\
&= \mathbb{E}_0\left[\frac{(y^4+4\mu_0^2y^2+\mu_0^4-4\mu_0y^3+2\mu_0^2y^2-4\mu_0^3y)}{4\sigma_0^4}\right] \\
&\quad+\mathbb{E}_0\left[\frac{(y^4+4\mu_1^2y^2+\mu_1^4-4\mu_1y^3+2\mu_1^2y^2-4\mu_1^3y)}{4\sigma_1^4}\right] \\
&\quad-\mathbb{E}_0\left[\frac{(y^4+4\mu_0\mu_1y^2+\mu_0^2\mu_1^2-2\mu_1y^3-2\mu_0y^3+\mu_0^2y^2+\mu_1^2y^2-2\mu_0\mu_1^2y-2\mu_0^2\mu_1y)}{2\sigma_0^2\sigma_1^2}\right], \\
&= \frac{\mathbb{E}_0[y^4]+4\mu_0^2\mathbb{E}_0[y^2]+\mu_0^4-4\mu_0\mathbb{E}_0[y^3]+2\mu_0^2\mathbb{E}_0[y^2]-4\mu_0^3\mathbb{E}_0[y]}{4\sigma_0^4} \\
&\quad+\frac{\mathbb{E}_0[y^4]+4\mu_1^2\mathbb{E}_0[y^2]+\mu_1^4-4\mu_1\mathbb{E}_0[y^3]+2\mu_1^2\mathbb{E}_0[y^2]-4\mu_1^3\mathbb{E}_0[y]}{4\sigma_1^4} \\
&\quad-\frac{\mathbb{E}_0[y^4]+4\mu_0\mu_1\mathbb{E}_0[y^2]+\mu_0^2\mu_1^2-2\mu_1\mathbb{E}_0[y^3]-2\mu_0\mathbb{E}_0[y^3]}{2\sigma_0^2\sigma_1^2} \\
&\quad+\frac{\mu_0^2\mathbb{E}_0[y^2]+\mu_1^2\mathbb{E}_0[y^2]-2\mu_0\mu_1^2\mathbb{E}_0[y]-2\mu_0^2\mu_1\mathbb{E}_0[y]}{2\sigma_0^2\sigma_1^2}, \\
&= \frac{\mu_0^4+6\mu_0^2\sigma_0^2+3\sigma_0^4+6\mu_0^4+6\mu_0^2\sigma_0^2-4\mu_0^4-12\mu_0^2\sigma_0^2-4\mu_0^4+\mu_0^4}{4\sigma_0^4} \\
&\quad+\frac{\mu_0^4+6\mu_0^2\sigma_0^2+3\sigma_0^4+6\mu_0^2\mu_1^2+6\mu_1^2\sigma_0^2-4\mu_0^3\mu_1-12\mu_0\mu_1\sigma_0^2-4\mu_0\mu_1^3+\mu_1^4}{4\sigma_1^4} \\
&\quad-\frac{\mu_0^4+6\mu_0^2\sigma_0^2+3\sigma_0^4+\left(\mu_0^2+\sigma_0^2\right)\left(4\mu_0\mu_1+\mu_0^2+\mu_1^2\right)-\left(\mu_0^3+3\mu_0\sigma_0^2\right)\left(2\mu_0+2\mu_1\right)}{2\sigma_0^2\sigma_1^2} \\
&\quad-\frac{\mu_0\left(2\mu_0\mu_1^2+2\mu_0^2\mu_1\right)+\mu_0^2\mu_1^2}{2\sigma_0^2\sigma_1^2}, \\
&= \frac{3}{4}+\frac{\mu_0^4+\mu_1^4-4\left(\mu_0^3\mu_1+\mu_0\mu_1^3\right)+6\left(\mu_0^2\sigma_0^2+\mu_1^2\sigma_0^2+\mu_0^2\mu_1^2-2\mu_0\mu_1\sigma_0^2\right)+3\sigma_0^4}{4\sigma_1^4} \\
&\quad-\frac{\mu_0^4+6\mu_0^2\sigma_0^2+3\sigma_0^4+4\mu_0^3\mu_1+\mu_0^4+\mu_0^2\mu_1^2+4\mu_0\mu_1\sigma_0^2+\mu_0^2\sigma_0^2+\mu_1^2\sigma_0^2}{2\sigma_0^2\sigma_1^2} \\
&\quad-\frac{-2\mu_0^4-2\mu_0^3\mu_1-6\mu_0^2\sigma_0^2-6\mu_0\mu_1\sigma_0^2-2\mu_0^2\mu_1^2-2\mu_0^3\mu_1+\mu_0^2\mu_1^2}{2\sigma_0^2\sigma_1^2}, \\
&= \frac{3}{4}+\frac{\mu_0^4+\mu_1^4-4\left(\mu_0^3\mu_1+\mu_0\mu_1^3\right)+6\left(\mu_0^2\sigma_0^2+\mu_1^2\sigma_0^2+\mu_0^2\mu_1^2-2\mu_0\mu_1\sigma_0^2\right)+3\sigma_0^4}{4\sigma_1^4} \\
&\quad-\frac{3\sigma_0^2-2\mu_0\mu_1+\mu_0^2+\mu_1^2}{2\sigma_1^2}.
\end{aligned}
$$

$$
\begin{aligned}
\sigma_{\eta,0}^2 &= \mathbb{E}_0[\eta^2] - \mu_{\eta,0}^2, \\
&= Z_3 + Z_2^2 + 2Z_1Z_2 - \left(Z_1^2 + Z_2^2 + 2Z_1Z_2\right), \\
&= Z_3 - \frac{\left(\mu_0^2 + \mu_1^2 - 2\mu_0\mu_1 + \sigma_0^2 - \sigma_1^2\right)^2}{4\sigma_1^4}, \\
&= Z_3 - \frac{\mu_0^4 + \mu_1^4 + 4\mu_0^2\mu_1^2 + \sigma_0^4 + \sigma_1^4 + 2\mu_0^2\mu_1^2 - 4\mu_0^3\mu_1 + 2\mu_0^2\sigma_0^2 - 2\mu_0^2\sigma_1^2 - 4\mu_0\mu_1^3}{4\sigma_1^4} \\
&\quad - \frac{2\mu_1^2\sigma_0^2 - 2\mu_1^2\sigma_1^2 - 4\mu_0\mu_1\sigma_0^2 + 4\mu_0\mu_1\sigma_1^2 - 2\sigma_0^2\sigma_1^2}{4\sigma_1^4}, \\
&= \frac{3}{4} + \frac{4\sigma_0^2\left(\mu_0 - \mu_1\right)^2 + 2\sigma_0^4 - \sigma_1^4 + 2\sigma_1^2\left(\mu_0 - \mu_1\right)^2 + 2\sigma_0^2\sigma_1^2}{4\sigma_1^4} - \frac{3\sigma_0^2 + \left(\mu_0 - \mu_1\right)^2}{2\sigma_1^2}, \\
&= \frac{1}{2}\left(1 + \frac{\sigma_0^4}{\sigma_1^4}\right) + \left(\mu_0 - \mu_1\right)^2 \frac{\sigma_0^2}{\sigma_1^4} - \frac{\sigma_0^2}{\sigma_1^2}.
\end{aligned}
$$

The mean and the variance of the log-likelihood ratio under the alternative hypothesis are given by

$$
\begin{aligned}
\mu_{\eta,1} &= \frac{\mu_0^2 + \mu_1^2 - 2\mu_0\mu_1 + \sigma_1^2 - \sigma_0^2}{2\sigma_0^2} + \log\left(\frac{\sigma_0}{\sigma_1}\right), \\
\sigma_{\eta,1}^2 &= \frac{1}{2}\left(1 + \frac{\sigma_1^4}{\sigma_0^4}\right) + \left(\mu_0 - \mu_1\right)^2 \frac{\sigma_1^2}{\sigma_0^4} - \frac{\sigma_1^2}{\sigma_0^2}.
\end{aligned}
$$

A.2 Decision Thresholds for the $\mathcal{CI}$SPRT

According to [Sahu & Kar, 2016], the probability of false alarm can be written as

$$
\begin{aligned}
P_{\text{FA}} &= P_0(S_k(T) \geq \upsilon), \\
&\leq \sum_{t=1}^{\infty} P_0(S_k(t) \geq \upsilon), \\
&\leq \sum_{t=1}^{\infty} \mathcal{Q}\left(\frac{\upsilon - \mu_{\eta,0}t}{\sigma_{\eta,0}\sqrt{\xi t}}\right).
\end{aligned}
$$

Using the property $\mathcal{Q}(x) \leq \frac{1}{2}e^{-\frac{x^2}{2}}$ and following the derivation in [Sahu & Kar, 2016], one obtains

$$
\begin{aligned}
P_{\text{FA}} &\leq \frac{1}{2}\sum_{t=1}^{\infty} e^{\frac{-\upsilon^2-\mu_{\eta,0}^2 t^2+2\upsilon\mu_{\eta,0}t}{2\sigma_{\eta,0}^2\xi t}}, \\
&= \frac{1}{2}e^{\frac{\upsilon\mu_{\eta,0}}{\sigma_{\eta,0}^2\xi}}\sum_{t=1}^{\infty} e^{\frac{-\upsilon^2-\mu_{\eta,0}^2 t^2}{2\sigma_{\eta,0}^2\xi t}}, \\
&= \frac{1}{2}e^{\frac{\upsilon\mu_{\eta,0}}{\sigma_{\eta,0}^2\xi}}\Bigg[\sum_{t=1}^{\lfloor\frac{\upsilon}{2\mu_{\eta,0}}\rfloor} e^{\frac{-\upsilon^2-\mu_{\eta,0}^2 t^2}{2\sigma_{\eta,0}^2\xi t}} + \sum_{t=\lfloor\frac{\upsilon}{2\mu_{\eta,0}}\rfloor+1}^{\lfloor\frac{\upsilon}{\mu_{\eta,0}}\rfloor} e^{\frac{-\upsilon^2-\mu_{\eta,0}^2 t^2}{2\sigma_{\eta,0}^2\xi t}} \\
&\qquad + \sum_{t=\lfloor\frac{\upsilon}{\mu_{\eta,0}}\rfloor+1}^{\lfloor\frac{2\upsilon}{\mu_{\eta,0}}\rfloor} e^{\frac{-\upsilon^2-\mu_{\eta,0}^2 t^2}{2\sigma_{\eta,0}^2\xi t}} + \sum_{t=\lfloor\frac{2\upsilon}{\mu_{\eta,0}}\rfloor+1}^{\infty} e^{\frac{-\upsilon^2-\mu_{\eta,0}^2 t^2}{2\sigma_{\eta,0}^2\xi t}}\Bigg], \\
&\leq \frac{1}{2}e^{\frac{\upsilon\mu_{\eta,0}}{\sigma_{\eta,0}^2\xi}}\Bigg[e^{-\frac{\upsilon\mu_{\eta,0}}{\sigma_{\eta,0}^2\xi}}\sum_{t=1}^{\lfloor\frac{\upsilon}{2\mu_{\eta,0}}\rfloor} e^{-\frac{\mu_{\eta,0}^2 t}{2\sigma_{\eta,0}^2\xi}} + e^{-\frac{\upsilon\mu_{\eta,0}}{2\sigma_{\eta,0}^2\xi}}\sum_{t=\lfloor\frac{\upsilon}{2\mu_{\eta,0}}\rfloor+1}^{\lfloor\frac{\upsilon}{\mu_{\eta,0}}\rfloor} e^{-\frac{\mu_{\eta,0}^2 t}{2\sigma_{\eta,0}^2\xi}} \\
&\qquad + e^{-\frac{\upsilon\mu_{\eta,0}}{4\sigma_{\eta,0}^2\xi}}\sum_{t=\lfloor\frac{\upsilon}{\mu_{\eta,0}}\rfloor+1}^{\lfloor\frac{2\upsilon}{\mu_{\eta,0}}\rfloor} e^{-\frac{\mu_{\eta,0}^2 t}{2\sigma_{\eta,0}^2\xi}} + \sum_{t=\lfloor\frac{2\upsilon}{\mu_{\eta,0}}\rfloor+1}^{\infty} e^{-\frac{\mu_{\eta,0}^2 t}{2\sigma_{\eta,0}^2\xi}}\Bigg].
\end{aligned}
$$

Approximating the sums above with infinite geometric series as in [Sahu & Kar, 2016] and using the relation

$$
\sum_{t=0}^{\infty} ar^t = \frac{a}{1-r}, \qquad \text{for } |r| < 1,
$$

leads to an upper bound on the probability of false alarm according to

$$
\begin{aligned}
P_{\text{FA}} &\leq \frac{1}{2}\frac{e^{\frac{\upsilon\mu_{\eta,0}}{\sigma^2_{\eta,0}\xi}}}{1-e^{-\frac{\mu^2_{\eta,0}}{2\sigma^2_{\eta,0}\xi}}}\left[e^{-\frac{\upsilon\mu_{\eta,0}}{\sigma^2_{\eta,0}\xi}}+e^{-\frac{\upsilon\mu_{\eta,0}}{2\sigma^2_{\eta,0}\xi}}e^{-\frac{\upsilon\mu_{\eta,0}}{4\sigma^2_{\eta,0}\xi}}+e^{-\frac{\upsilon\mu_{\eta,0}}{4\sigma^2_{\eta,0}\xi}}e^{-\frac{\upsilon\mu_{\eta,0}}{2\sigma^2_{\eta,0}\xi}}+e^{-\frac{\upsilon\mu_{\eta,0}}{\sigma^2_{\eta,0}\xi}}\right],\\
&\leq \frac{e^{\frac{4\upsilon\mu_{\eta,0}}{4\sigma^2_{\eta,0}\xi}}}{1-e^{-\frac{\mu^2_{\eta,0}}{2\sigma^2_{\eta,0}\xi}}}\left[e^{-\frac{4\upsilon\mu_{\eta,0}}{4\sigma^2_{\eta,0}\xi}}+e^{-\frac{3\upsilon\mu_{\eta,0}}{4\sigma^2_{\eta,0}\xi}}\right],\\
&\leq \frac{2e^{\frac{\upsilon\mu_{\eta,0}}{4\sigma^2_{\eta,0}\xi}}}{1-e^{-\frac{\mu^2_{\eta,0}}{2\sigma^2_{\eta,0}\xi}}}.
\end{aligned}
$$

Requiring $P_{\text{FA}} \leq \alpha$ and solving for υ yields the upper threshold υ as

$$
\begin{aligned}
\alpha &\leq \frac{2e^{\frac{\upsilon\mu_{\eta,0}}{4\sigma^2_{\eta,0}\xi}}}{1-e^{-\frac{\mu^2_{\eta,0}}{2\sigma^2_{\eta,0}\xi}}},\\
\frac{\alpha}{2}\left(1-e^{-\frac{\mu^2_{\eta,0}}{2\sigma^2_{\eta,0}\xi}}\right) &\leq e^{\frac{\upsilon\mu_{\eta,0}}{4\sigma^2_{\eta,0}\xi}},\\
\log\left(\frac{\alpha}{2}\right)+\log\left(1-e^{-\frac{\mu^2_{\eta,0}}{2\sigma^2_{\eta,0}\xi}}\right) &\leq \frac{\upsilon\mu_{\eta,0}}{4\sigma^2_{\eta,0}\xi},\\
\upsilon &\geq \frac{4\sigma^2_{\eta,0}\xi}{\mu_{\eta,0}}\left[\log\left(\frac{\alpha}{2}\right)+\log\left(1-e^{-\frac{\mu^2_{\eta,0}}{2\sigma^2_{\eta,0}\xi}}\right)\right].
\end{aligned}
$$

Repeating the same procedure for the probability of misdetection and requiring $P_{\text{MD}} \leq \beta$ yields the lower threshold

$$
\lambda \leq \frac{4\sigma^2_{\eta,1}\xi}{\mu_{\eta,1}}\left[\log\left(\frac{\beta}{2}\right)+\log\left(1-e^{-\frac{\mu^2_{\eta,1}}{2\sigma^2_{\eta,1}\xi}}\right)\right].
$$

B

Appendix to Part II

B.1 Pseudo-Code of the Diffusion Particle PHD Filter

input: $d, E_c, K, n, N, N_{\mathsf{p}}, p_{\mathsf{B}}, p_{\mathsf{S}}, \lambda_{\mathsf{FA}}, c_{\mathsf{FA}}$

initialize: $\left\{\boldsymbol{s}^p_{k,\mathsf{coll}}(0), w^p_{k,\mathsf{coll}}(0)\right\}_{p=1}^{N_{k,\mathsf{coll}}(0)} = \left\{\boldsymbol{s}^p_{k,\mathsf{new}}(0), w^p_{k,\mathsf{new}}(0)\right\}_{p=1}^{N_{k,\mathsf{new}}(0)} = \emptyset.$

while $t \leq n$ **do**

  for $k = 1, \ldots, N$ **do**

    Merge sets of collective persistent and newborn particles with weights:

$$\left\{\boldsymbol{s}^p_{k,\mathsf{tot}}(t), w^p_{k,\mathsf{tot}}(t)\right\}_{p=1}^{N_{k,\mathsf{tot}}(t)} = \left\{\boldsymbol{s}^p_{k,\mathsf{coll}}(t-1), w^p_{k,\mathsf{coll}}(t-1)\right\}_{p=1}^{N_{k,\mathsf{coll}}(t-1)} \cup \left\{\boldsymbol{s}^p_{k,\mathsf{new}}(t-1), w^p_{k,\mathsf{new}}(t-1)\right\}_{p=1}^{N_{k,\mathsf{new}}(t-1)}.$$

    for $p = 1, \ldots, N_{k,\mathsf{tot}}(t)$ **do**

      Predict new particle state and update weight:

$$\boldsymbol{s}^p_{k,\mathsf{pers}}(t) = \boldsymbol{F}\boldsymbol{s}^p_{k,\mathsf{tot}}(t),$$
$$w^p_{k,\mathsf{pers}}(t \,|\, t-1) = p_{\mathsf{S}} w^p_{k,\mathsf{tot}}(t).$$

      Update weights using neighborhood measurements:

$$w^p_{k,\mathtt{pers}}(t) = \prod_{l \in \mathcal{N}_k} \left[1 - p_D + \sum_{\boldsymbol{z}_j \in \Sigma^l_t} w^p_{k,j,\mathtt{update}}(t) \right] w^p_{k,\mathtt{pers}}(t \,|\, t-1),$$

$$w^p_{k,j,\mathtt{update}}(t) = \frac{p_D f_t(\boldsymbol{z}_j \mid \boldsymbol{x}^p(t))}{\lambda_{\mathtt{FA}} c_{\mathtt{FA}}(\boldsymbol{z}_j) + \mathcal{L}(\boldsymbol{z}_j)},$$

$$\mathcal{L}(\boldsymbol{z}_j) = \sum_{q=1}^{N_{k,\mathtt{pers}}(t)} p_D f_t(\boldsymbol{z}_j \mid \boldsymbol{x}^q(t)) w^q_{k,\mathtt{pers}}(t \,|\, t-1).$$

9: **end for**

10: Form set of candidate measurements for ATB:

$$\Sigma^k_{t,\mathtt{cand}} = \Sigma^k_t \backslash \Big\{ \boldsymbol{z}_{m_p} \,\Big|\, m_p = \arg\max_j \; w^p_{k,j,\mathtt{update}}(t), \quad p = 1, \ldots, N_{k,\mathtt{pers}}(t), \quad j = 1, \ldots, \left|\Sigma^l_t\right| \quad \forall \; l \in \mathcal{N}_k \Big\}.$$

11: Calculate estimated number of targets:

$$\hat{N}_{k,\mathtt{tgt}}(t) = \left\lfloor \sum_{p=1}^{N_{k,\mathtt{tot}}(t)} w^p_{k,\mathtt{pers}}(t) \right\rceil.$$

12: Resample $\hat{N}_{k,\mathtt{pers}}(t) = \hat{N}_{k,\mathtt{tgt}}(t) N_{\mathtt{p}}$ particles and reset weights:

$$w^p_{k,\mathtt{pers}}(t) = \frac{\hat{N}_{k,\mathtt{tgt}}(t)}{N_{k,\mathtt{pers}}(t)}, \qquad p = 1, \ldots, N_{k,\mathtt{pers}}(t).$$

13: Merge sets of persistent neighborhood particles and weights:

$$\left\{ \boldsymbol{s}^p_{k,\mathtt{coll}}(t), w^p_{k,\mathtt{coll}}(t) \right\}_{p=1}^{N_{k,\mathtt{coll}}(t)} = \bigcup_{l \in \mathcal{N}_k} \left\{ \boldsymbol{s}^p_{l,\mathtt{pers}}(t), w^p_{l,\mathtt{pers}}(t) \right\}_{p=1}^{N_{l,\mathtt{pers}}(t)}.$$

14: Use single-linkage clustering to identify $\hat{N}_{\mathtt{tgt}}(t)$ clusters.

15: Find set of estimated target states $\left\{ \hat{\boldsymbol{s}}^l_k(t) \right\}_{l=1}^{\hat{N}_{\mathtt{tgt}}(t)}$ by calculating centroids.

16: Add independent jitter to each particle with standard deviation:

$$\sigma^j_c(t) = K E_c N_{k,\mathtt{coll}}(t)^{-1/d}.$$

17: Place $N_{\mathtt{p}}$ particles randomly around each candidate measurement $\boldsymbol{z}_j \in \Sigma^k_{t,\mathtt{cand}}$.

Set weights according to:

$$w^p_{k,\mathsf{new}}(t) = \frac{p_\mathsf{B}}{N_{k,\mathsf{new}}(t)}, \qquad p = 1, \dots, N_{k,\mathsf{new}}(t).$$

end for

$t \leftarrow t + 1$

end while

return

B.2 Pseudo-Code of the Multi-Sensor Particle PHD Filter

input: $d, E_c, K, n, N, N_\mathsf{p}, p_\mathsf{B}, p_\mathsf{S}, \lambda_\mathsf{FA}, c_\mathsf{FA}$

initialize: $\{\boldsymbol{s}^p_\mathsf{coll}(0), w^p_\mathsf{coll}(0)\}_{p=1}^{N_\mathsf{coll}(0)} = \{\boldsymbol{s}^p_\mathsf{new}(0), w^p_\mathsf{new}(0)\}_{p=1}^{N_\mathsf{new}(0)} = \emptyset$.

while $t \leq n$ **do**

Merge sets of persistent and newborn particles with weights:

$$\{\boldsymbol{s}^p_\mathsf{tot}(t), w^p_\mathsf{tot}(t)\}_{p=1}^{N_\mathsf{tot}(t)} = \{\boldsymbol{s}^p_\mathsf{pers}(t-1), w^p_\mathsf{pers}(t-1)\}_{p=1}^{N_\mathsf{pers}(t-1)} \cup \{\boldsymbol{s}^p_\mathsf{new}(t-1), w^p_\mathsf{new}(t-1)\}_{p=1}^{N_\mathsf{new}(t-1)}.$$

for $p = 1, \dots, N_\mathsf{tot}(t)$ **do**

Predict new particle state and update weight:

$$\boldsymbol{s}^p_\mathsf{pers}(t) = \boldsymbol{F}\boldsymbol{s}^p_\mathsf{tot}(t),$$
$$w^p_\mathsf{pers}(t \,|\, t-1) = p_\mathsf{S} w^p_\mathsf{tot}(t).$$

Cluster measurements using single-linkage clustering.

Assign each measurement a label reflecting the cluster cardinality.

Update weights using measurements of the entire network:

$$w^p_\mathsf{pers}(t) = \left[1 - p_D + \sum_{\boldsymbol{z}_j \in \Sigma_t} w^p_{j,\mathsf{update}}(t)\right] w^p_\mathsf{pers}(t \,|\, t-1),$$
$$w^p_{j,\mathsf{update}}(t) = \frac{p_D f_t(\boldsymbol{z}_j \mid \boldsymbol{x}^p(t))}{(\lambda_\mathsf{FA} c_\mathsf{FA}(\boldsymbol{z}_j) + \mathcal{L}(\boldsymbol{z}_j))\, C(\boldsymbol{z}_j)},$$
$$\mathcal{L}(\boldsymbol{z}_j) = \sum_{q=1}^{N_\mathsf{pers}(t)} p_D f_t(\boldsymbol{z}_j \mid \boldsymbol{x}^q(t)) w^q_\mathsf{pers}(t \,|\, t-1).$$

end for

Form the set of candidate measurements for ATB:

$$\Sigma_{t,\mathsf{cand}} = \Sigma_t \backslash \left\{ \boldsymbol{z}_{m_p} \;\middle|\; m_p = \arg\max_j \; w^p_{j,\mathsf{update}}(t), \quad p = 1, ..., N_{\mathsf{pers}}(t), \quad j = 1, ..., |\Sigma_t| \right\}.$$

Calculate the estimated number of targets:

$$\hat{N}_{\mathsf{tgt}}(t) = \left\lfloor \sum_{p=1}^{N_{\mathsf{tot}}(t)} w^p_{\mathsf{pers}}(t) \right\rfloor.$$

Resample $\hat{N}_{\mathsf{pers}}(t) = \hat{N}_{\mathsf{tgt}}(t) N_{\mathsf{p}}$ particles and reset weights:

$$w^p_{\mathsf{pers}}(t) = \frac{\hat{N}_{\mathsf{tgt}}(t)}{N_{\mathsf{pers}}(t)}, \qquad p = 1, \ldots, N_{\mathsf{pers}}(t).$$

Apply k-means clustering.

Calculate cluster centroids, form set of estimated target states $\left\{ \hat{\boldsymbol{s}}^l(t) \right\}_{l=1}^{\hat{N}_{\mathsf{tgt}}(t)}$

Add independent jitter to each particle with standard deviation:

$$\sigma^j_c(t) = K E_c N_{\mathsf{pers}}(t)^{-1/d}.$$

Place N_{p} particles randomly around each candidate measurement $\boldsymbol{z}_j \in \Sigma_{t,\mathsf{cand}}$.

Set weights as:

$$w^p_{\mathsf{new}}(t) = \frac{p_{\mathsf{B}}}{N_{\mathsf{new}}(t)}, \qquad p = 1, \ldots, N_{\mathsf{new}}(t).$$

$t \leftarrow t + 1$

end while

return

List of Abbreviations & Acronyms

ARL	average runlength
ATB	adaptive target birth
BPA	basic probability assignment
BPA-CDF	basic probability assignment based on the cumulative distribution function of the measurement
BPA-LLR	basic probability assignment based on the cumulative distribution function of the log-likelihood ratio
BPA-PDF	basic probability assignment based on the probability density function of the measurement
CDF	cumulative distribution function
$\mathcal{CI}$SPRT	Consensus+Innovations Sequential Probability Ratio Test
$\mathcal{CI}$MSPRT	Consensus+Innovations Matrix Sequential Probability Ratio Test
DDF-PPHDF	Distributed Data Fusion Particle PHD Filter
D-PPHDF	Diffusion Particle PHD Filter
DPCRLB	Distributed Posterior Cramér-Rao Lower Bound
DST	Dempster-Shafer Theory of Evidence
LFDs	least favorable densities
LFD-$\mathcal{CI}$SPRT	Least-Favorable-Density Consensus+Innovations Sequential Probability Ratio Test
LFD-$\mathcal{CI}$MSPRT	Least-Favorable-Density Consensus+Innovations Matrix Sequential Probability Ratio Test
LLR	log-likelihood ratio
M-$\mathcal{CI}$SPRT	M-estimator Consensus+Innovations Sequential Probability Ratio Test
M-$\mathcal{CI}$MSPRT	M-estimator Consensus+Innovations Matrix Sequential Probability Ratio Test

M-D-PPHDF	M-estimator Diffusion Particle PHD Filter
Median-$\mathcal{CI}$SPRT	Median Consensus+Innovations Sequential Probability Ratio Test
Median-$\mathcal{CI}$MSPRT	Median Consensus+Innovations Matrix Sequential Probability Ratio Test
Median-D-PPHDF	Median Diffusion Particle PHD Filter
MS-PPHDF	Multi-Sensor Particle Probability Hypothesis Density Filter
MSPRT	Matrix Sequential Probability Ratio Test
Myriad-$\mathcal{CI}$SPRT	Myriad Consensus+Innovations Sequential Probability Ratio Test
Myriad-$\mathcal{CI}$MSPRT	Myriad Consensus+Innovations Matrix Sequential Probability Ratio Test
OSPA	Optimal Subpattern Assignment
PCRLB	Posterior Cramér-Rao Lower Bound
PDF	probability density function
PHD	Probability Hypothesis Density
SPRT	Sequential Probability Ratio Test

List of Figures

List of Notations & Symbols

The following list contains the most important notations and symbols in the dissertation. The remaining symbols are introduced upon usage.

Notations

$\check{\cdot}, \hat{\cdot}$	robust / estimated quantity
$(\cdot)^\top$	transposition
$\lvert \cdot \rvert$	absolute value / cardinality of a set
$\lVert \cdot \rVert_2, \lVert \cdot \rVert_{\max}$	Euclidean distance / maximum norm of a matrix
$\lfloor \cdot \rceil$	nearest-integer function
$\mathbb{E}[\cdot], \mathbb{E}_m[\cdot]$	expected value (under $\mathcal{H}_m$)

Arabic Symbols

$\mathbf{0}_{N,M}$	zero matrix (vector) of size $N \times M$
$\mathbf{1}$	one-vector
2^Ω	power set
$b_t(\boldsymbol{s})$	PHD of birth set $\boldsymbol{\mathcal{B}}_t$
$B_{t,\text{dist}}$	Distributed PCRLB at time t
$Bel(A)$	belief for event A
$c_{\text{FA}}(\boldsymbol{z})$	PDF of false alarms / clutter
c_{Hub}	constant for tuning Huber's M-estimator
$D(p_n \mid p_m)$	Kullback-Leibler divergence between p_n and p_m
$D_{t\mid t}(\boldsymbol{s}(t) \mid \Sigma_{0:t})$	PHD of state $\boldsymbol{s}(t)$ given measurements $\Sigma_{0:t}$
$\boldsymbol{e}_k$	kth column of identity matrix $\boldsymbol{I}$

E_c	empirical interval length for jitter
$\boldsymbol{F}, \boldsymbol{G}$	matrix describing the linear target motion model / state noise model
H, h	contamination distribution and its probability density
$\boldsymbol{H}_k$	measurement matrix of node k
$\boldsymbol{I}$	identity matrix
K	tuning constant for roughening
$m(A)$	probability mass assigned to event A
$m_k(t, \mathcal{H}_m)$	probability mass assigned to event A at node k under $\mathcal{H}_m$ at time t
$\boldsymbol{n}^{\text{tgt}}$	state noise vector
N	number of nodes
$N_{\text{coll}}, N_{\text{new}}, N_{\text{pers}}$	number of collective / new / persistent particles
N_{p}	number of particles per target
N_{tot}	total number of particles
$N_{k,\text{coll}}, N_{k,\text{new}}, N_{k,\text{pers}}$	number of collective / new / persistent particles at node k
$N_{k,\text{tot}}$	total number of particles at node k
N_{MC}	number of Monte Carlo runs
$\hat{N}^{\text{tgt}}, \hat{N}_k^{\text{tgt}}$	estimated number of targets (at node k)
p, p°, p_m	(nominal) probability density function (under $\mathcal{H}_m$)
$p_{\text{B}}, p_{\text{D}}, p_{\text{S}}$	probability of target birth / detection / survival
P, P°, P_m	(nominal) probability distribution (under $\mathcal{H}_m$)
$P_{\text{FA}}, P_{\text{MD}}$	probability of false alarm / misdetection
$Pl(A)$	plausability for event A
q, q_m	least favorable density (under $\mathcal{H}_m$)
$\boldsymbol{Q}$	covariance matrix of Gaussian state noise
$r_{\text{com}}, r_{\text{sen}}$	communication / sensing radius
$\boldsymbol{R}_k$	measurement noise covariance matrix of node k
$\boldsymbol{s}^{\text{tgt}}$	target state vector
$\boldsymbol{s}^p$	state vector of particle p
$\boldsymbol{s}^p_{\text{coll}}, \boldsymbol{s}^p_{\text{new}}, \boldsymbol{s}^p_{\text{pers}}$	state vector of collective / new / persistent particle p
$\boldsymbol{s}^p_{\text{tot}}$	state vector of total particle p
$\boldsymbol{s}^p_k$	state vector of particle p of node k

$\boldsymbol{s}^p_{k,\text{coll}}$, $\boldsymbol{s}^p_{k,\text{new}}$, $\boldsymbol{s}^p_{k,\text{pers}}$	state vector of collective / new / persistent particle p at node k
$\boldsymbol{s}^p_{k,\text{tot}}$	state vector of total particle p at node k
$S(t)$, $\check{S}_k(t)$, $S_k(t)$	(robust) test statistic (of node k) at time t
$S_{mn}(t)$, $S^k_{mn}(t)$	pairwise test statistic (of node k) at time t between $\mathcal{H}_m/\mathcal{H}_n$
T	stopping time
w^p	weight of particle p
w^p_{coll}, w^p_{new}, w^p_{pers}	weight of collective / new / persistent particle p
w^p_{tot}	weight of total particle p
w^p_k	weight of particle p at node k
$w^p_{k,\text{coll}}$, $w^p_{k,\text{new}}$, $w^p_{k,\text{pers}}$	weight of collective / new / persistent particle p at node k
$w^p_{k,\text{tot}}$	weight of total particle p at node k
$\boldsymbol{W}$	weighting matrix
$\hat{x}^{\text{mean}}$, $\hat{x}^{\text{median}}$, $\hat{x}^{\text{myriad}}$	sample mean / median / myriad of x
$\hat{x}^{\text{M}}$	M-estimate of x
$\boldsymbol{x}_k$, $\boldsymbol{x}^{\text{tgt}}$	location vector of node k / target location vector
$y_k(t)$, $Y_k(t)$	measurement of node k at time t and corresponding random variable
$\boldsymbol{z}_k$	target measurement vector of node k

Calligraphic Symbols

$\boldsymbol{\mathcal{B}}_t$	birth set of new targets entering the scene at time t
$\boldsymbol{\mathcal{C}}_t$	random finite set of clutter or false alarms at time t
$\mathcal{G}(\mathcal{V},\mathcal{E})$	graph of nodes $\mathcal{V}$ and edges $\mathcal{E}$
$\mathcal{H}_m$	hypothesis m
$\mathcal{N}(\mu,\sigma^2)$	normal distribution with mean μ and variance σ^2
$\mathcal{N}_k$, $\mathcal{N}^o_k$	closed / open neighborhood of node k
$\mathcal{P}_m$	set of probability distributions under $\mathcal{H}_m$
$\mathcal{Q}(x)$	right tail-probability of the standard normal distribution
$\boldsymbol{\mathcal{S}}_t$	survival set of targets surviving the transition from $t-1$ to t

Greek Symbols

α, α_{mn}	bound on probability of false alarm (for pairwise test between $\mathcal{H}_m/\mathcal{H}_n$)
β, β_{mn}	bound on probability of misdetection (for pairwise test between $\mathcal{H}_m/\mathcal{H}_n$)
δ	decision (and stopping) rule
ε	contamination factor
η_k, η_k^{clipped}	(clipped) LLR of node k
η_{mn}^k, $\eta_{mn}^{k,\text{clipped}}$	(clipped) LLR of node k between $\mathcal{H}_m/\mathcal{H}_n$
$\boldsymbol{\eta}$	vector of LLRs
κ	constant for weighting the noise power
λ, $\check{\lambda}$, λ_{mn}	(robust) lower decision threshold (for pairwise test between $\mathcal{H}_m/\mathcal{H}_n$)
λ_{FA}	average number of false alarms / clutter
μ, μ_m, $\mu_{\eta,m}$	mean / mean (of the LLR) under $\mathcal{H}_m$
Ω	frame of discernment
$\boldsymbol{\nu}_k^{\text{tgt}}$	measurement noise of node k
$\psi(x)$	score function of the M-estimator
ρ_k	reliability of node k
σ, $\hat{\sigma}_{\text{mad}}$	(median) standard deviation
σ_c^j	componentwise standard deviation of the jitter for roughening
σ^2, σ_m^2, $\sigma_{\eta,m}^2$	variance / variance (of the LLR) under $\mathcal{H}_m$
σ_q^2, σ_r^2	componentwise power of the measurement noise / state noise
ς_k, $\varsigma_{k,l}$	similarity of the belief of node k to that of its neighbors / of node l
$\boldsymbol{\varsigma}$	similarity vector
$\boldsymbol{\Sigma}$, Σ	random finite set of multitarget measurements and its realization
$\Sigma_{0:t}$	set of all multitarget measurements from time 0 to t
$\Sigma_{t,\text{cand}}^k$	set of candidate measurements for ATB at node k and time t
Σ_t	set of all target measurements at time t
$\boldsymbol{\Theta}_t(\Xi)$	random finite set of measurements at time t
υ, $\check{\upsilon}$, υ_{mn}	(robust) upper decision threshold (for pairwise test between $\mathcal{H}_m/\mathcal{H}_n$)
ξ	constant for tuning the upper bound of the variance of $S_k(t)$
$\boldsymbol{\Xi}$, Ξ	random finite set of multi-target states and its realization

References

[Akyildiz et al., 2002] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam & E. Cayirci. "A Survey on Sensor Networks." In: *IEEE Communications Magazine* 40.8 (Aug. 2002), pp. 102–114 (cited on page 10).

[Al-Sayed et al., 2017] S. Al-Sayed, A. M. Zoubir & A. H. Sayed. "Robust Distributed Estimation by Networked Agents." In: *IEEE Transactions on Signal Processing* 65.15 (Aug. 2017), pp. 3909–3921 (cited on page 2).

[Arulampalam et al., 2002] M. Arulampalam, S. Maskell, N. Gordon & T. Clapp. "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking." In: *IEEE Transactions on Signal Processing* 50.2 (2002), pp. 174–188 (cited on page 3).

[Balthasar et al., 2014] M. R. Balthasar, S. Al-Sayed, S. Leier & A. M. Zoubir. "Optimal Area Coverage in Autonomous Sensor Networks." In: *Proceedings of the 2nd International Conference and Exhibition on Underwater Acoustics (UA2014)*. Invited Paper. June 2014 (cited on pages 1, 91).

[Bar-Shalom et al., 2009] Y. Bar-Shalom, F. Daum & J. Huang. "The Probabilistic Data Association Filter." In: *IEEE Control Systems Magazine* 29.6 (2009), pp. 82–100 (cited on page 84).

[Bar-Shalom et al., 2011] Y. Bar-Shalom, P. K. Willett & X. Tian. *Tracking and Data Fusion: A Handbook of Algorithms*. Storrs, Connecticut, USA: YBS Publishing, 2011 (cited on pages 3, 103).

[Battistelli et al., 2013] G. Battistelli, L. Chisci, C. Fantacci, A. Farina & A. Graziano. "Consensus CPHD Filter for Distributed Multitarget Tracking." In: *IEEE Journal of Selected Top-*

ics in Signal Processing 7.3 (2013), pp. 508–520 (cited on pages 3, 124).

[Bessell et al., 2003] A. Bessell, B. Ristic, A. Farina, X. Wang & M. Arulampalam. "Error Performance Bounds for Tracking a Manoeuvring Target." In: *Proceedings of the 6th International Conference of Information Fusion (FUSION)*. vol. 2. 2003, pp. 903–910 (cited on pages 111, 113).

[Blum et al., 1997] R. S. Blum, S. A. Kassam & H. V. Poor. "Distributed Detection with Multiple Sensors II. Advanced Topics." In: *Proceedings of the IEEE* 85.1 (1997), pp. 64–79 (cited on page 2).

[Braca et al., 2012] P. Braca, S. Marano, V. Matta & P. Willett. "Multitarget-Multisensor ML and PHD: Some Asymptotics." In: *Proceedings of the 15th International Conference on Information Fusion (FUSION)*. 2012, pp. 2347–2353 (cited on page 110).

[Braca et al., 2013] P. Braca, S. Marano, V. Matta & P. Willett. "Asymptotic Efficiency of the PHD in Multitarget/Multisensor Estimation." In: *IEEE Journal of Selected Topics in Signal Processing* 7.3 (June 2013), pp. 553–564 (cited on pages 110, 113).

[Cattivelli & Sayed, 2011] F. S. Cattivelli & A. H. Sayed. "Distributed Detection over Adaptive Networks using Diffusion Adaptation." In: *IEEE Transactions on Signal Processing* 59.5 (2011), pp. 1917–1932 (cited on page 1).

[Cattivelli et al., 2008] F. S. Cattivelli, C. G. Lopes & A. H. Sayed. "Diffusion Strategies for Distributed Kalman Filtering: Formulation and Performance Analysis." In: *Proceedings of the 1st IAPR Workshop on Cognitive Information Processing (CIP)* (2008), pp. 36–41 (cited on page 3).

[Challa et al., 2011] S. Challa, R. Evans, M. Morelande & D. Musicki. *Fundamentals of Object Tracking*. Cambridge, UK: Cambridge University Press, 2011 (cited on pages 3, 86, 87).

[Clark, 2006] D. E. Clark. "Multiple Target Tracking with the Probability Hypothesis Density Filter." Doctoral disserta-

tion. Edinburgh, UK: Heriot-Watt University, Oct. 2006 (cited on pages 3, 84, 89, 93, 103, 124).

[DeGroot, 1960] M. H. DeGroot. "Minimax Sequential Tests of Some Composite Hypotheses." In: *The Annals of Mathematical Statistics* (1960), pp. 1193–1200 (cited on page 2).

[Everitt et al., 2011] B. Everitt, S. Landau, M. Leese & D. Stahl. *Cluster Analysis*. Chichester, UK: John Wiley & Sons, Ltd., 2011 (cited on pages 97, 104).

[Fauß & Zoubir, 2015] M. Fauß & A. M. Zoubir. "A Linear Programming Approach to Sequential Hypothesis Testing." In: *Sequential Analysis* 34.2 (2015), pp. 235–263 (cited on page 2).

[Fauß & Zoubir, 2016] M. Fauß & A. M. Zoubir. "Old Bands, New Tracks—Revisiting the Band Model for Robust Hypothesis Testing." In: *IEEE Transactions on Signal Processing* 64.22 (Aug. 2016), pp. 5875–5886 (cited on pages 2, 17, 22, 23).

[Fauß, 2016] M. Fauß. "Design and Analysis of Optimal and Minimax Robust Sequential Hypothesis Tests." Doctoral dissertation. Darmstadt, Germany: Technische Universität Darmstadt, June 2016 (cited on pages 2, 23).

[Gonzalez & Arce, 1996] J. G. Gonzalez & G. R. Arce. "Weighted Myriad Filters: A Robust Filtering Framework Derived from Alpha-Stable Distributions." In: *Proceedings of the 21st IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. vol. 5. May 1996, pp. 2833–2836 (cited on page 10).

[Gonzalez & Arce, 2002] J. G. Gonzalez & G. R. Arce. "Statistically-Efficient Filtering in Impulsive Environments: Weighted Myriad Filters." In: *EURASIP Journal on Advances in Signal Processing*. 2002, pp. 4–20 (cited on page 10).

[Gordon et al., 1993] N. J. Gordon, D. J. Salmond & A. F. M. Smith. "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation." In: *IEE Proceedings F – Radar and Signal Processing*. Vol. 140. 2. 1993, pp. 107–113 (cited on page 97).

[Granstrom & Orguner, 2012] K. Granstrom & U. Orguner. "A PHD Filter for Tracking Multiple Extended Targets Using Random Matrices." In: *IEEE Transactions on Signal Processing* 60.11 (Nov. 2012), pp. 5657–5671 (cited on page 104).

[Granstrom et al., 2014] K. Granstrom, A. Natale, P. Braca, G. Ludeno & F. Serafino. "PHD Extended Target Tracking using an Incoherent X-Band Radar: Preliminary Real-World Experimental Results." In: *Proceedings of the 17th International Conference on Information Fusion (FUSION)*. July 2014, pp. 1–8 (cited on page 104).

[Gül & Zoubir, 2017a] G. Gül & A. M. Zoubir. "Minimax Robust Hypothesis Testing." In: *IEEE Transactions on Information Theory* (2017) (cited on pages 1, 2).

[Gül & Zoubir, 2017b] G. Gül & A. M. Zoubir. "Theoretical Bounds in Minimax Decentralized Hypothesis Testing." In: *IEEE Transactions on Signal Processing* 65.1 (2017), pp. 15–26 (cited on page 2).

[Gül, 2017] G. Gül. *Robust and Distributed Hypothesis Testing*. Vol. 414. Cham, Switzerland: Springer International Publishing AG, 2017 (cited on page 2).

[Gustafsson, 2010] F. Gustafsson. "Particle Filter Theory and Practice with Positioning Applications." In: *IEEE Aerospace and Electronic Systems Magazine* 25.7 (2010), pp. 53–82 (cited on pages 99, 107).

[Gustafsson et al., 2002] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson & P.-J. Nordlund. "Particle Filters for Positioning, Navigation, and Tracking." In: *IEEE Transactions on Signal Processing* 50.2 (2002), pp. 425–437 (cited on pages 84, 85).

[Han et al., 2012] Y. Han, Q. Chen & J. Wang. "An Enhanced D-S Theory Cooperative Spectrum Sensing Algorithm against SSDF Attack." In: *Proceedings of the 75th IEEE Vehicular Technology Conference (VTC Spring)*. May 2012, pp. 1–5 (cited on page 73).

[Hlinka et al., 2013] O. Hlinka, F. Hlawatsch & P. M. Djuric. "Distributed Particle Filtering in Agent Networks: A Survey, Classification, and Comparison." In: *IEEE Signal Processing Magazine* 30.1 (2013), pp. 61–81 (cited on page 3).

[Hoffmann & Mahler, 2004] J. Hoffman & R. Mahler. "Multitarget Miss Distance via Optimal Assignment." In: *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 34.3 (2004), pp. 327–336 (cited on page 110).

[Hong et al., 2011] S. Hong, L. Wang, Z.-G. Shi & K. S. Chen. "Simplified Particle PHD Filter for Multiple-Target Tracking: Algorithm and Architecture." In: *Progress In Electromagnetics Research* 120 (2011), pp. 481–498 (cited on pages 93, 103).

[Hou, Leonard et al., 2017] W. Hou, M. R. Leonard & A. M. Zoubir. "Robust Distributed Sequential Detection via Robust Estimation." In: *Proceedings of the 25th European Signal Processing Conference (EUSIPCO)*. Sept. 2017 (cited on pages 25, 28, 42, 43).

[Huber & Strassen, 1973] P. J. Huber & V. Strassen. "Minimax Tests and the Neyman-Pearson Lemma for Capacities." In: *The Annals of Statistics* (1973), pp. 251–263 (cited on page 1).

[Huber, 1964] P. J. Huber. "Robust Estimation of a Location Parameter." In: *The Annals of Mathematical Statistics* 35.1 (1964), pp. 73–101 (cited on page 1).

[Huber, 1965] P. J. Huber. "A Robust Version of the Probability Ratio Test." In: *The Annals of Mathematical Statistics* 36.6 (1965), pp. 1753–1758 (cited on pages 1, 23).

[Huber, 1981] P. J. Huber. *Robust Statistics*. Hoboken, New Jersey, USA: Wiley, 1981 (cited on pages 1, 9, 23).

[Hue et al., 2002a] C. Hue, J.-P. Le Cadre & P. Perez. "Performance Analysis of Two Sequential Monte Carlo Methods and Posterior Cramer-Rao Bounds for Multi-Target Tracking." In: *Proceedings of the 5th International Conference on Information Fusion (FUSION)*. vol. 1. 2002, pp. 464–473 (cited on page 111).

[Hue et al., 2002b] C. Hue, J.-P. Le Cadre & P. Perez. "Sequential Monte Carlo Methods for Multiple Target Tracking and Data Fusion." In: *IEEE Transactions on Signal Processing* 50.2 (2002), pp. 309–325 (cited on page 111).

[Hue et al., 2002c] C. Hue, J.-P. Le Cadre & P. Pérez. *Performance Analysis of Two Sequential Monte Carlo Methods and Posterior Cramér-Rao Bounds for Multi-Target Tracking*. Tech. rep. IRISA, 2002 (cited on page 111).

[Inaba et al., 1994] M. Inaba, N. Katoh & H. Imai. "Applications of Weighted Voronoi Diagrams and Randomization to Variance-based K-clustering: (Extended Abstract)." In: *Proceedings of the 10th Annual Symposium on Computational Geometry (SCG)*. 1994, pp. 332–339 (cited on page 107).

[Kalman, 1960] R. E. Kalman. "A New Approach to Linear Filtering and Prediction Problems." In: *Journal of Basic Engineering* 82.1 (1960), pp. 35–45 (cited on page 88).

[Kar & Moura, 2013] S. Kar & J. M. F. Moura. "Consensus + Innovations Distributed Inference over Networks: Cooperation and Sensing in Networked Systems." In: *IEEE Signal Processing Magazine* 30.3 (May 2013), pp. 99–109 (cited on page 27).

[Karl & Willig, 2007] H. Karl & A. Willig. *Protocols and Architectures for Wireless Sensor Networks*. Chichester, UK: John Wiley & Sons, Ltd., 2007 (cited on page 10).

[Kassam, 1981] S. A. Kassam. "Robust Hypothesis Testing for Bounded Classes of Probability Densities." In: *IEEE Transactions on Information Theory* 27 (1981), pp. 242–247 (cited on page 22).

[Kay, 1993] S. M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, New Jersey, USA: Prentice-Hall, Inc., 1993 (cited on page 110).

[Kay, 2006] S. M. Kay. *Intuitive Probability and Random Processes Using MATLAB®*. New York City, New York, USA:

Springer Science & Business Media, LLC, 2006 (cited on page 38).

[Lehmann & Romano, 2005] E. L. Lehmann & J. P. Romano. *Testing Statistical Hypotheses*. New York City, New York, USA: Springer Science & Business Media, LLC, 2005 (cited on page 38).

[Leonard & Zoubir, 2017] M. R. Leonard & A. M. Zoubir. "Robust Distributed Sequential Hypothesis Testing for Detecting a Random Signal in Non-Gaussian Noise." In: *Proceedings of the 25th European Signal Processing Conference (EUSIPCO)*. Sept. 2017 (cited on pages 25, 28, 43).

[Leonard & Zoubir, 2018a] M. R. Leonard & A. M. Zoubir. "Robust Sequential Detection in Distributed Sensor Networks." In: *IEEE Transactions on Signal Processing* 66.21 (Nov. 2018), pp. 5648–5662 (cited on pages 11, 25).

[Leonard & Zoubir, 2019] M. R. Leonard & A. M. Zoubir. "Multi-Target Tracking in Distributed Sensor Networks using Particle PHD Filters." In: *Signal Processing* 159 (June 2019), pp. 130–146 (cited on pages 1, 83).

[Leonard et al., 2018a] M. R. Leonard, C. A. Schroth & A. M. Zoubir. "Dempster-Shafer Theory Based Robust Sequential Detection in Distributed Sensor Networks." In: *Proceedings of the IEEE Statistical Signal Processing Workshop (SSP)*. June 2018 (cited on page 66).

[Leonard et al., 2018b] M. R. Leonard, M. Stiefel, M. Fauß & A. M. Zoubir. "Robust Sequential Testing of Multiple Hypotheses in Distributed Sensor Networks." In: *Proceedings of the 43nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Mar. 2018 (cited on page 51).

[Leonard et al., 2018c] M. R. Leonard, M. Stiefel, M. Fauß & A. M. Zoubir. "Robustifying Sequential Multiple Hypothesis Tests in Distributed Sensor Networks." In: *Proceedings of the 26th European Signal Processing Conference (EUSIPCO)*. Sept. 2018 (cited on page 52).

[Levy, 2008] B. C. Levy. *Principles of Signal Detection and Parameter Estimation*. New York City, New York, USA: Springer Science & Business Media, LLC, 2008 (cited on pages 1, 17, 18, 23).

[Li & Wang, 2018] S. Li & X. Wang. "Fully Distributed Sequential Hypothesis Testing: Algorithms and Asymptotic Analyses." In: *IEEE Transactions on Information Theory* 64.4 (2018), pp. 2742–2758 (cited on page 2).

[Liu & Mei, 2017] K. Liu & Y. Mei. "Improved Performance Properties of the CISPRT Algorithm for Distributed Sequential Detection." ONLINE: https://www2.isye.gatech.edu/~ymei/papers/p_LiuMei_IEEE-SP2017.pdf (cited on page 2).

[MacQueen, 1967] J. MacQueen. "Some Methods for Classification and Analysis of Multivariate Observations." In: *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*. Vol. 1. 281-297. 1967, p. 14 (cited on page 106).

[Mahler, 2001] R. Mahler. *Multitarget Moments and Their Application to Multitarget Tracking*. Tech. rep. DTIC, 2001 (cited on pages 88, 89).

[Mahler, 2003] R. Mahler. "Multitarget Bayes Filtering via First-Order Multitarget Moments." In: *IEEE Transactions on Aerospace and Electronic Systems* 39.4 (2003), pp. 1152–1178 (cited on pages 3, 84, 86, 88, 90).

[Mahler, 2007] R. Mahler. *Statistical Multisource-Multitarget Information Fusion*. Norwood, Massachusetts, USA: Artech House, Inc., 2007 (cited on page 86).

[Mahler, 2009] R. Mahler. "The Multisensor PHD Filter: I. General Solution via Multitarget Calculus." In: *Proceedings of the SPIE Signal Processing, Sensor Fusion, and Target Recognition XVIII*. vol. 7336. May 2009 (cited on page 124).

[Mahler, 2010] R. Mahler. "Approximate Multisensor CPHD and PHD Filters." In: *Proceedings of the 13th Interna-*

tional Conference on Information Fusion (FUSION). July 2010, pp. 1–8 (cited on page 124).

[Mahler, 2013] R. Mahler. "'Statistics 102' for Multisource-Multitarget Detection and Tracking." In: *IEEE Journal of Selected Topics in Signal Processing* 7.3 (2013), pp. 376–389 (cited on page 88).

[Mahler, 2014] R. Mahler. *Advances in Statistical Multisource-Multitarget Information Fusion*. Artech House, Inc., 2014 (cited on page 86).

[Maresca et al., 2014] S. Maresca, P. Braca, J. Horstmann & R. Grasso. "Maritime Surveillance Using Multiple High-Frequency Surface-Wave Radars." In: *IEEE Transactions on Geoscience and Remote Sensing* 52.8 (Aug. 2014), pp. 5056–5071 (cited on page 3).

[Matta et al., 2016] V. Matta, P. Braca, S. Marano & A. H. Sayed. "Distributed Detection over Adaptive Networks: Refined Asymptotics and the Role of Connectivity." In: *IEEE Transactions on Signal and Information Processing over Networks* 2.4 (2016), pp. 442–460 (cited on page 1).

[Murtagh, 1983] F. Murtagh. "A Survey of Recent Advances in Hierarchical Clustering Algorithms." In: *The Computer Journal* 26.4 (1983), pp. 354–359 (cited on pages 100, 107).

[Neyman & Pearson, 1933] J. Neyman & E. S. Pearson. "On the Problem of the Most Efficient Tests of Statistical Hypotheses." In: *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 231.694-706 (1933), pp. 289–337 (cited on page 19).

[Nguyen-Thanh & Koo, 2009] N. Nguyen-Thanh & I. Koo. "An Enhanced Cooperative Spectrum Sensing Scheme Based on Evidence Theory and Reliability Source Evaluation in Cognitive Radio Context." In: *IEEE Communications Letters* 13.7 (2009), pp. 492–494 (cited on page 71).

[Novikov, 2009a] A. Novikov. "Optimal Sequential Multiple Hypothesis Tests." In: *Kybernetika* 45.2 (2009), pp. 309–330 (cited on page 2).

[Novikov, 2009b] A. Novikov. "Optimal Sequential Tests for Two Simple Hypotheses." In: *Sequential Analysis* 28.2 (2009), pp. 188–217 (cited on page 2).

[Oh et al., 2007] S. Oh, L. Schenato, P. Chen & S. Sastry. "Tracking and Coordination of Multiple Agents Using Sensor Networks: System Design, Algorithms and Experiments." In: *Proceedings of the IEEE* 95.1 (2007), pp. 234–254 (cited on pages 3, 84).

[Olfati-Saber et al., 2007] R. Olfati-Saber, J. A. Fax & R. M. Murray. "Consensus and Cooperation in Networked Multi-Agent Systems." In: *Proceedings of the IEEE* 95.1 (Jan. 2007), pp. 215–233 (cited on pages 3, 11).

[Poor, 2013] H. V. Poor. *An Introduction to Signal Detection and Estimation*. New York City, New York, USA: Springer Science & Business Media, LLC, 2013 (cited on pages 17, 23).

[Qihang et al., 2006] P. Qihang, Z. Kun, W. Jun & L. Shaoqian. "A Distributed Spectrum Sensing Scheme Based on Credibility and Evidence Theory in Cognitive Radio Context." In: *Proceedings of the 17th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*. 2006, pp. 1–5 (cited on page 69).

[Rambach et al., 2015] J. Rambach, M. F. Huber, M. R. Balthasar & A. M. Zoubir. "Collaborative Multi-Camera Face Recognition and Tracking." In: *Proceedings of the 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. Aug. 2015 (cited on page 3).

[Reid, 1979] D. Reid. "An Algorithm for Tracking Multiple Targets." In: *IEEE Transactions on Automatic Control* 24.6 (1979), pp. 843–854 (cited on pages 3, 84).

[Ristic et al., 2010] B. Ristic, D. Clark & B.-N. Vo. "Improved SMC Implementation of the PHD Filter." In: *Proceedings of the 13th International Conference on Information Fusion (FUSION)*. July 2010, pp. 1–8 (cited on page 93).

[Ristic et al., 2012] B. Ristic, D. Clark, B.-N. Vo & B.-T. Vo. "Adaptive Target Birth Intensity for PHD and CPHD Filters." In: *IEEE Transactions on Aerospace and Electronic Systems* 48.2 (Apr. 2012), pp. 1656–1668 (cited on pages 92, 93).

[Sahu & Kar, 2014] A. K. Sahu & S. Kar. "Distributed Sequential Detection for Gaussian Binary Hypothesis Testing: Heterogeneous Networks." In: *Proceedings of the 48th Asilomar Conference on Signals, Systems and Computers*. Nov. 2014, pp. 723–727 (cited on pages 2, 4, 27, 79).

[Sahu & Kar, 2016] A. K. Sahu & S. Kar. "Distributed Sequential Detection for Gaussian Shift-in-Mean Hypothesis Testing." In: *IEEE Transactions on Signal Processing* 64.1 (Jan. 2016), pp. 89–103 (cited on pages 2, 4, 11, 27, 28, 32–34, 42, 43, 132, 133).

[Sayed, 2013] A. H. Sayed. "Diffusion Adaptation over Networks." In: *E-Reference Signal Processing*. Ed. by R. Chellappa & S. Theodoridis. New York City, New York, USA: Elsevier, 2013 (cited on pages 1, 28, 93).

[Schmitz, 1987] N. Schmitz. "Minimax Sequential Tests of Composite Hypotheses on the Drift of a Wiener Process." In: *Statistical Papers* 28.1 (1987), pp. 247–261 (cited on page 2).

[Schuhmacher et al., 2008] D. Schuhmacher, B.-T. Vo & B.-N. Vo. "A Consistent Metric for Performance Evaluation of Multi-Object Filters." In: *IEEE Transactions on Signal Processing* 56.8 (Aug. 2008), pp. 3447–3457 (cited on page 110).

[Shafer, 1976] G. Shafer. *A Mathematical Theory of Evidence*. Princeton, New Jersey, USA: Princeton University Press, 1976 (cited on pages 65, 66, 79).

[Tartakovsky et al., 2014] A. Tartakovsky, I. Nikiforov & M. Basseville. *Sequential Analysis: Hypothesis Testing and Changepoint Detection*. Boca Raton, Florida, USA: CRC Press, 2014 (cited on pages 1, 2, 51, 52).

[Teneketzis & Ho, 1987] D. Teneketzis & Y.-C. Ho. "The Decentralized Wald Problem." In: *Information and Computation* 73.1 (1987), pp. 23–44 (cited on page 2).

[Tichavsky et al., 1998] P. Tichavsky, C. Muravchik & A. Nehorai. "Posterior Cramer-Rao Bounds for Discrete-Time Nonlinear Filtering." In: *IEEE Transactions on Signal Processing* 46.5 (1998), pp. 1386–1396 (cited on page 111).

[Tu & Sayed, 2011] S.-Y. Tu & A. H. Sayed. "Mobile Adaptive Networks." In: *IEEE Journal of Selected Topics in Signal Processing* 5.4 (2011), pp. 649–664 (cited on page 1).

[Uney et al., 2010] M. Uney, S. Julier, D. Clark & B. Ristic. "Monte Carlo Realisation of a Distributed Multi-Object Fusion Algorithm." In: *Proceedings of the 1st Conference on Sensor Signal Processing for Defence (SSPD)*. Sept. 2010, pp. 1–5 (cited on pages 3, 109, 124).

[Uney et al., 2013] M. Uney, D. Clark & S. Julier. "Distributed Fusion of PHD Filters Via Exponential Mixture Densities." In: *IEEE Journal of Selected Topics in Signal Processing* 7.3 (June 2013), pp. 521–531 (cited on pages 3, 124).

[Van Trees, 2004] H. L. Van Trees. *Detection, Estimation, and Modulation Theory*. Chichester, UK: John Wiley & Sons, Ltd., 2004 (cited on pages 17, 111).

[Veeravalli, 1999] V. V. Veeravalli. "Sequential Decision Fusion: Theory and Applications." In: *Journal of the Franklin Institute* 336.2 (1999), pp. 301–322 (cited on page 11).

[Veeravalli et al., 1993] V. V. Veeravalli, T. Basar & H. V. Poor. "Decentralized Sequential Detection with a Fusion Center Performing the Sequential Test." In: *IEEE Transactions on Information Theory* 39.2 (Mar. 1993), pp. 433–442 (cited on page 11).

[Veeravalli et al., 1994] V. V. Veeravalli, T. Basar & H. V. Poor. "Minimax Robust Decentralized Detection." In: *IEEE Transactions on Information Theory* 40.1 (1994), pp. 35–40 (cited on page 2).

[Vo & Ma, 2006] B.-N. Vo & W.-K. Ma. "The Gaussian Mixture Probability Hypothesis Density Filter." In: *IEEE Transactions on Signal Processing* 54.11 (2006), pp. 4091–4104 (cited on pages 100, 124).

[Vo, 2008] B.-T. Vo. "Random Finite Sets in Multi-Object Filtering." Doctoral dissertation. University of Western Australia, 2008 (cited on page 86).

[Vo et al., 2003] B.-N. Vo, S. Singh & A. Doucet. "Sequential Monte Carlo Implementation of the PHD filter for Multi-Target Tracking." In: *Proceedings of the 6th International Conference on Information Fusion (FUSION)*. 2003, pp. 792–799 (cited on pages 86, 93, 103).

[Vo et al., 2005] B.-N. Vo, S. Singh & A. Doucet. "Sequential Monte Carlo Methods for Multitarget Filtering with Random Finite Sets." In: *IEEE Transactions on Aerospace and Electronic Systems* 41.4 (2005), pp. 1224–1245 (cited on pages 86, 87, 124).

[Wald, 1945] A. Wald. "Sequential Tests of Statistical Hypotheses." In: *The Annals of Mathematical Statistics* 16.2 (1945), pp. 117–186 (cited on page 2).

[Wald, 1947] A. Wald. *Sequential Analysis*. New York City, New York, USA: John Wiley & Sons, Ltd., 1947 (cited on pages 2, 17, 20, 27).

[Wang et al., 2003] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless & C. Gill. "Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks." In: *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*. ACM. 2003, pp. 28–39 (cited on page 91).

[Wang et al., 2014] J. Wang, S. Feng, Q. Wu, X. Zheng, Y. Xu & G. Ding. "A Robust Cooperative Spectrum Sensing Scheme Based on Dempster-Shafer Theory and Trustworthiness Degree Calculation in Cognitive Radio Networks." In: *EURASIP Journal on Advances in Signal Processing* 1 (2014), p. 201 (cited on page 73).

[Xiao & Boyd, 2004] L. Xiao & S. Boyd. "Fast Linear Iterations for Distributed Averaging." In: *Systems & Control Letters* 53.1 (2004), pp. 65–78 (cited on page 27).

[Zoubir et al., 2012] A. M. Zoubir, V. Koivunen, Y. Chakhchoukh & M. Muma. "Robust Estimation in Signal Processing: A

Tutorial-Style Treatment of Fundamental Concepts." In: *IEEE Signal Processing Magazine* 29.4 (July 2012), pp. 61–80 (cited on pages 1, 7–9).

[Zoubir et al., 2018] A. M. Zoubir, V. Koivunen, E. Ollila & M. Muma. *Robust Statistics for Signal Processing*. Cambridge, UK: Cambridge University Press, 2018 (cited on pages 1, 7, 9).

Erklärungen laut Promotionsordnung

§ 8 Abs. 1 lit. c PromO

Ich versichere hiermit, dass die elektronische Version meiner Dissertation mit der schriftlichen Version übereinstimmt.

§ 8 Abs. 1 lit. d PromO

Ich versichere hiermit, dass zu einem vorherigen Zeitpunkt noch keine Promotion versucht wurde. In diesem Fall sind nähere Angaben über Zeitpunkt, Hochschule, Dissertationsthema und Ergebnis dieses Versuchs mitzuteilen.

§ 9 Abs. 1 PromO

Ich versichere hiermit, dass die vorliegende Dissertation selbstständig und nur unter Verwendung der angegebenen Quellen verfasst wurde.

§ 9 Abs. 2 PromO

Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

Darmstadt, 04. Dezember 2018

Mark Ryan Leonard

THIS THESIS WAS TYPESET using LaTeX, originally developed by Leslie Lamport and based on Donald Knuth's TeX. The body text is set in 12 point Egenolff-Berner Garamond, a revival of Claude Garamont's humanist typeface. A template that can be used to format a PhD dissertation with this look & feel has been released under the permissive AGPL license, and can be found online at github.com/suchow/Dissertate or from its lead author, Jordan Suchow, at suchow@post.harvard.edu.