Distributed Estimation Using Partial Knowledge about Correlated Estimation Errors

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Susanne Radtke
aus Ilmenau

Tag der mündlichen Prüfung:	11.02.2022
Erster Gutachter:	Prof. Dr.-Ing. Uwe D. Hanebeck
Zweiter Gutachter:	doc. Ing. Ondřej Straka, Ph.D.

Acknowledgment

This thesis is a result of my research performed at the Intelligent Sensor-Actuator-Systems (ISAS) laboratory, Institute for Anthropomatics and Robotics (IAR), Karlsruhe Institute of Technology (KIT). First, I am deeply grateful to my advisor, Uwe D. Hanebeck, for his support and the opportunities he has offered me. Without his guidance and patience, this thesis would not have been possible. Second, I would also like to thank my second advisor, Ondřej Straka, for his valuable insights and suggestions on how to improve my work.

My research was part of two Joint German-Czech research projects with the University of West Bohemia, funded by the German Research Foundation (DFG). I also enjoyed participating in the Software Campus initiative, funded by the Federal Ministry of Education and Research (BMBF). Therefore, I also want to thank everyone at Software Campus and Trumpf GmbH + Co. KG for the opportunity to learn and grow.

I had the pleasure of working with many wonderful people over the past couple of years, and I am grateful to everyone who accompanied me through my research's ups and downs. First, I want to thank Benjamin Noack for sparking my interest in distributed estimation and for being a great mentor. Furthermore, I want to thank Jiří Ajgl for his insights and patient explanations. Furthermore, many thanks to all of my (former) colleagues: Ajit Basarur, Michael Fennel, Daniel Frisch, Christopher Funk, Kailai Li, Jana Mayer, Lukas Michiels, Florian Pfaff, Marcel Reith-Braun, Marko Ristic, Florian Rosenthal, Selim Özgen, Johannes Westermann, and Antonio Zea, for the coffee breaks, exciting discussion, adventures during our conference trips, and making work enjoyable. Furthermore, I want to thank our technician team, Alexander Riffel, Sascha Faber, Achim Langendörfer, and our secretary Dagmar Gambichler, for supporting my daily work.

I also want to voice my appreciation to all family members who supported me along the way, particularly my mother, Angela. Furthermore, I want to extend my thanks to my friends, especially Dominik, Ella, Henning, Jens, Martin, and Rene, for lending their ears, putting things into perspective, and making me smile. Last but not least, I want to thank my partner, Vale, for the love and support that helped me get through all the difficulties and worries.

Thanks, everyone!

Hamburg, July 2022 Susanne Radtke

Contents

Notation V

Zusammenfassung VII

Abstract IX

1 Introduction 1

1.1 Context 1

1.2 Problem Formulation 3

1.3 Related Work 4

1.4 Open Challenges of Data Fusion in Sensor Networks 5

1.5 Contributions and Outline 6

2 Distributed Estimation 9

2.1 Distributed State Estimation in Sensor Networks 9

2.1.1 Local Estimation 10

2.1.2 Communication between Sensor Nodes 11

2.1.3 Fusion of State Estimates 13

2.2 Correlation Uncertainty in Distributed Estimation 17

2.2.1 Reasons and Types of Correlation Uncertainty 18

2.2.2 Approaches for Correlation Retrieval 19

2.3 Conclusions to Distributed Estimation 20

3 Full Reconstruction of Cross-Covariances 21

3.1 Optimal Centralized Reconstruction of Cross-covariances 22

3.2 Distributed Track-keeping of Cross-Covariances 23

3.2.1 Reconstruction Based on Square Root Decomposition 24

3.2.2 Deterministic Sample-Based Reconstruction 25

3.2.3 Comparison Between the Square Root Decomposition and the Sample-Based Reconstruction 27
3.3 Track-Keeping for Different Local Coordinate Systems 28
3.3.1 Linearly Transformed State Spaces 30
3.3.2 Overlapping State Estimates 36
3.3.3 Conclusion to Reconstruction of Cross-Covariances for Estimates from Different Coordinate Systems 40
3.4 Fully Decentralized Track-Keeping of Cross-Covariances 40
3.4.1 Optimal Reconstruction for Hierarchical Network Topologies 42
3.4.2 Optimal Reconstruction for Fully Decentralized Network Topologies 43
3.4.3 Conclusion to Fully Decentralized Track-Keeping of Cross-Covariances . . 45
3.5 Conclusions to Full Reconstruction of Cross-Covariances 45

4 Partial Reconstruction of Cross-Covariances 47
4.1 Related Work 47
4.2 Partial Reconstruction of Cross-Covariances 48
4.2.1 Reducing the Amount of Tracked Information 49
4.2.2 Extension to Fully Decentralized Network Topologies 55
4.3 Conclusions to Partial Reconstruction of Cross-Covariances 62

5 Learning Partial Knowledge about Correlation 63
5.1 The Cross-Correlation Matrix 63
5.2 Estimation of Uncertain Correlations 67
5.2.1 Sample Correlation 67
5.2.2 Bayesian Estimation of the Correlation Matrix 69
5.3 Conclusions to Learning Partial Knowledge about Correlation 73

6 Exploiting Partial Knowledge about Correlation 75
6.1 Methods for Information Retrieval 76
6.2 Methods to Exploit Partial Knowledge 78
6.2.1 Bounding Methods Using Partial Knowledge 78
6.2.2 A Gaussian Mixture Approach to Fusion 80
6.3 Parameter Identification for Design of Conservative Bounds 86

6.3.1 Designing Bounds for the Analytic Approach 86
6.3.2 Designing Bounds for the Simulation Approach 89
6.3.3 Discussion 90
6.4 Evaluation 90
6.4.1 Evaluation of the Simulation Approach 91
6.4.2 Evaluation of the Analytic Approach 93
6.5 Conclusions to Exploiting Partial Knowledge about Correlation 94

7 Conclusions and Future Research **97**
7.1 Key Contributions 97
7.2 Relevance 99
7.3 Future Work 99

List of Figures and Tables **101**

Bibliography **103**

Supervised Student Theses **113**

Own Publications **115**

Notation

General Conventions

x	scalar value
$\underline{x}$	vector
$\mathbf{A}$	matrix
$\mathbf{A}^{\mathrm{T}}$	transpose of a matrix
$\mathbf{A}^{-1}$	inverse of a matrix
$\sqrt{\mathbf{A}}$	cholesky decomposition of a matrix
$\mathrm{diag}(\mathbf{A})$	vector with diagonal entries of a matrix
$\mathrm{blkdiag}(\mathbf{A}_1, \mathbf{A}_2)$	matrix with block-diagonal entries $\mathbf{A}_1$ and $\mathbf{A}_2$
$\mathrm{tr}(\mathbf{A})$	trace of a matrix
svd	singular value decomposition
$\mathbf{I}$	identity matrix
$\underline{0}$	vector where all entries are zero
$\mathbf{0}$	matrix where all entries are zero
$\{\cdot\}_{m=1}^{M}$	set with M entries
$(\cdot)^i$	quantity of sensor node i
$(\cdot)_k$	quantity at time step k
$\underline{\mathbf{x}}$	vector containing random values
$\underline{\tilde{\mathbf{x}}}$	estimation error of $\underline{\mathbf{x}}$
$\mathrm{E}(\underline{x})$	expected value of $\underline{x}$
$\mathcal{N}(\underline{x}, \mathbf{P})$	Gaussian distribution with mean $\underline{x}$ and covariance $\mathbf{P}$
$\mathcal{E}_{\Theta,\underline{\mu}}$	ellipse with covariance Θ and center $\underline{\mu}$

State Estimation

$\underline{\hat{x}}_{k\|k-1}$	predicted state estimate at time step k
$\underline{\hat{x}}_{k\|k}$	estimated state estimate at time step k
$\mathbf{P}_{k\|k-1}$	predicted covariance matrix at time step k
$\mathbf{P}_{k\|k}$	estimated covariance matrix at time step k
$\mathbf{A}$	system matrix
$\mathbf{B}$	input matrix
$\mathbf{C}$	measurement matrix
$\mathbf{Q}$	covariance matrix of process noise
$\mathbf{R}^i$	measurement noise covariance matrix of sensor node i
$\mathbf{K}$	Kalman filter gain for measurement
$\mathbf{L}$	Kalman filter gain for predicted estimate

Distributed Estimation

$\mathbf{J}$	joint covariance matrix
$\mathbf{H}$	measurement matrix for the joint state space
$(\cdot)^\mathrm{f}$	fused quantity
$\mathbf{F}$	fusion gain
$\mathbf{\Sigma}_\mathbf{Q}$	square root decomposition of process noise covariance
$\mathbf{\Sigma}_{\mathbf{R}^i}$	square root decomposition of measurement noise covariance of node i
$\mathbf{S}_\mathbf{Q}$	square root matrix of process noise covariances
$\mathbf{S}_{\mathbf{R}^i}$	square root matrix of measurement noise covariances of node i
$\mathcal{T}$	time horizon

Correlation Coefficients and Matrices

ϱ	correlation coefficient
$\mathbf{\Lambda}$	correlation matrix
c	sample correlation
$\mathbf{D}$, $\mathbf{M}$	centering matrix, normalized centering matrix
ρ	largest singular value
$\mathbf{\Psi}$, $\mathbf{\Phi}$	scaling matrix, normalized scaling matrix
$\underline{\lambda}$	parameter vector

Abbreviations

ANEES	Averaged Normalized Estimation Error Squared
BSC	Bar-Shalom–Campo
BLUE	Best Linear Unbiased Estimator
CI	Covariance Intersection
EKF	Extended Kalman Filter
ICI	Inverse Covariance Intersection
IFM	Information Matrix Fusion
LMMSE	Linear Minimum Mean Square Error
MCR	Monte Carlo Runs
ML	Maximum Likelihood
MSE	Mean Squared Error
pdf	Probability density function
SCI	Split Covariance Intersection
SIS	Sequential Importance Sampling
T2TF	Track-to-Track Fusion
UKF	Unscented Kalman Filter
WLS	Weighted Least Squares

Zusammenfassung

Sensornetzwerke werden in vielen verschiedenen Anwendungen, z. B. zur Überwachung des Flugraumes oder zur Lokalisierung in Innenräumen eingesetzt. Dabei werden Sensoren häufig räumlich verteilt, um eine möglichst gute Abdeckung des zu beobachtenden Prozesses zu ermöglichen. Sowohl der Prozess als auch die Sensormessungen unterliegen stochastischem Rauschen. Daher wird oftmals eine Zustandsschätzung, z. B. durch ein Kalmanfilter durchgeführt, welcher die Unsicherheiten aus dem Prozess- und Messmodel systematisch berücksichtigt. Die Kooperation der individuellen Sensorknoten erlaubt eine verbesserte Schätzung des Systemzustandes des beobachteten Prozesses. Durch die lokale Verarbeitung der Sensordaten direkt in den Sensorknoten können Sensornetzwerke flexibel und modular entworfen werden und skalieren auch bei steigender Anzahl der Einzelkomponenten gut. Zusätzlich werden Sensornetzwerke dadurch robuster, da die Funktionsfähigkeit des Systems nicht von einem einzigen zentralen Knoten abhängt, der alle Sensordaten sammelt und verarbeitet. Ein Nachteil der verteilten Schätzung ist jedoch die Entstehung von korrelierten Schätzfehlern durch die lokale Verarbeitung in den Filtern. Diese Korrelationen müssen systematisch berücksichtigt werden, um genau und zuverlässig den Systemzustand zu schätzen. Dabei muss oftmals ein Kompromiss zwischen Schätzgenauigkeit und den begrenzt verfügbaren Ressourcen wie Bandbreite, Speicher und Energie gefunden werden. Eine zusätzliche Herausforderung sind unterschiedliche Netzwerktopologien sowie die Heterogenität lokaler Informationen und Filter, welche das Nachvollziehen der individuellen Verarbeitungsschritte innerhalb der Sensorknoten und der korrelierten Schätzfehler erschweren.

Diese Dissertation beschäftigt sich mit der Fusion von Zustandsschätzungen verteilter Sensorknoten. Speziell wird betrachtet, wie korrelierte Schätzfehler entweder vollständig oder teilweise gelernt werden können, um eine präzisere und weniger unsichere fusionierte Zustandsschätzung zu erhalten. Um Wissen über korrelierte Schätzfehler zu erhalten, werden in dieser Arbeit sowohl analytische als auch simulations-basierte Ansätze verfolgt.

Eine analytische Berechnung der Korrelationen zwischen Zustandsschätzungen ist möglich, wenn alle Verarbeitungsschritte und Parameter der lokalen Filter bekannt sind. Dadurch kann z. B. ein zentraler Fusionsknoten die die Korrelation zwischen den Schätzfehlern rekonstruieren. Dieses zentralisierte Vorgehen ist jedoch oft sehr aufwendig und benötigt entweder eine hohe Kommunikationsrate oder Vorwissen über die lokale Verarbeitungsschritte und Filterparameter. Daher wurden in den letzten Jahren zunehmend dezentrale Methoden zur Rekonstruktion von Korrelationen zwischen Zustandsschätzungen erforscht. In dieser Arbeit werden Methoden zur dezentralen Nachverfolgung und Rekonstruktion von korrelierten Schätzfehlern diskutiert und weiterentwickelt. Dabei basiert der erste Ansatz auf der Verwendung deterministischer Samples und der zweite auf der Wurzelzerlegung korrelierter Rauschkovarianzen. Um die Verwendbarkeit dieser Methoden zu steigern, werden mehrere wichtige Erweiterungen erarbeitet. Zum Einen schätzen verteilte Sensorknoten häufig den Zustand desselben Systems. Jedoch unterscheiden sie sich in ihrer lokalen Berechnung, indem sie unterschiedliche Zustandsraummodelle nutzen. Ein Beitrag dieser Arbeit ist daher die Verallgemeinerung dezentraler Methoden zur Nachverfolgung in unterschiedlichen (heterogenen) Zustandsräumen gleicher oder geringerer Dimension, die durch lineare Transformationen entstehen. Des Weiteren ist die Rekonstruktion begrenzt auf Systeme mit einem einzigen zentralen Fusionsknoten. Allerdings stellt die Abhängigkeit des Sensornetzwerkes von einem solchen zentralen Knoten einen Schwachpunkt dar, der im Fehlerfall zum vollständigen Ausfall des Netzes führen kann. Zudem verfügen viele Sensornetzwerke über komplexe und variierende Netzwerktopologien ohne zentralen Fusionsknoten. Daher ist eine weitere wichtige Errungenschaft dieser Dissertation die Erweiterung der Methodik auf die Rekonstruktion korrelierter Schätzfehler unabhängig von der genutzten Netzwerkstruktur. Ein Nachteil der erarbeiteten Algorithmen sind die wachsenden Anforderungen an Speicherung, Verarbeitung und Kommunikation der zusätzlichen Informationen, welche für die vollständige Rekonstruktion notwendig sind. Um diesen Mehraufwand zu begrenzen, wird ein Ansatz zur teilweisen Rekonstruktion korrelierter Schätzfehler erarbeitet Das resultierende partielle Wissen über korrelierte Schätzfehler benötigt eine konservative Abschätzung der Unsicherheit, um genaue und zuverlässige Zustandsschätzungen zu erhalten.

Es gibt jedoch Fälle, in denen keine Rekonstruktion der Korrelationen möglich ist oder es eine Menge an möglichen Korrelationen gibt. Dies ist zum Einen der Fall, wenn mehrere Systemmodelle möglich sind. Dies führt dann zu einer Menge möglicher korrelierter Schätzfehler, beispielsweise wenn die Anzahl der lokalen Verarbeitungsschritte bis zur Fusion ungewiss ist. Auf der anderen Seite ist eine Rekonstruktion auch nicht möglich, wenn die Systemparameter nicht bekannt sind oder die Rekonstruktion aufgrund von begrenzter Rechenleistung nicht ausgeführt werden kann. In diesem Fall kann ein Simulationsansatz verwendet werden, um die Korrelationen zu schätzen. In dieser Arbeit werden Ansätze zur Schätzung von Korrelationen zwischen Schätzfehlern basierend auf der Simulation des gesamten Systems erarbeitet. Des Weiteren werden Ansätze zur vollständigen und teilweisen Rekonstruktion einer Menge korrelierter Schätzfehler für mehrere mögliche Systemkonfigurationen entwickelt. Diese Mengen an Korrelationen benötigen entsprechende Berücksichtigung bei der Fusion der Zustandsschätzungen. Daher werden mehrere Ansätze zur konservativen Fusion analysiert und angewendet. Zuletzt wird ein Verfahren basierend auf Gaußmischdichten weiterentwickelt, dass die direkte Verwendung von Mengen an Korrelationen ermöglicht.

Die in dieser Dissertation erforschten Methoden bieten sowohl Nutzern als auch Herstellern von verteilten Schätzsystemen einen Baukasten an möglichen Lösungen zur systematischen Behandlung von korrelierten Schätzfehlern. Abhängig von der Art und den Umfang des Wissens über Korrelationen, der Kommunikationsbandbreite sowie der gewünschten Qualität der fusionierten Schätzung kann eine Methode passgenau aus den beschriebenen Methoden zusammengesetzt und angewendet werden. Die somit geschlossene Lücke in der Literatur eröffnet neue Möglichkeiten für verteilte Sensorsysteme in verschiedenen Anwendungsgebieten.

Abstract

Sensor networks are essential for many different applications, e.g., air space surveillance or indoor localization. Thereby, sensors are often spatially distributed to allow for better coverage for observing the process of interest. However, the process and the sensor measurements are subject to stochastic noise. Therefore, the state of the process is estimated, e.g., using a Kalman filter that systematically considers the uncertainties of the process and measurements. Sensor nodes can improve their estimation by cooperating with other nodes to execute their estimation tasks by exchanging local information among each other or a central processing unit. This makes the sensor network design more flexible and modular, and it scales better with an increasing number of sensor nodes. Additionally, the sensor networks are more robust because the failure of a single sensor node that collects and processes all measurements serves as a single point of failure. Unfortunately, the local estimation and the exchange of information between sensor nodes result in correlated estimation errors. These correlations have to be accounted for to ensure credible and reliable fusion results. Because of limited resources, e.g., bandwidth, memory, and energy, compromises between the quality of the estimation and costs have to be made. Additionally, sensor networks pose more challenges, e.g., different network topologies or heterogeneous local information and filters, making the reconstruction of individual processing steps and correlated estimation errors demanding.

This thesis focuses on the fusion of state estimates from distributed sensor nodes. In particular, the focal point is to retrieve either full or partial knowledge of correlated estimation errors that are subsequently used to increase the quality and credibility of the fused estimate. In order to obtain this knowledge, this thesis proposes an analytic and a simulation approach.

The analytic calculation of correlated estimation errors is possible when all processing steps and the parameters of the local estimators are known. In this case, e.g., a central fusion node can reconstruct the correlation of estimation errors. However, this centralized approach suffers from a high communication rate or requires previously determined knowledge about local processing steps and filtering parameters. To alleviate this limitation, distributed methods to keep track of correlated estimation errors have received increasing attention within recent years. This thesis introduces and further develops methods to keep track and reconstruct correlated estimation errors using deterministic samples and square root decompositions of correlated noise covariances. Several significant limitations of these approaches are addressed and solved in this thesis. The first limitation is that local estimators often observe the same phenomenon, but their local estimation can be carried out in different local coordinate systems. Therefore, an essential contribution of this thesis is the generalization to track keeping in systems with linearly transformed local state-spaces in the same or lower dimensions. Further, distributed track-keeping methods have been limited to reconstructing correlated estimation errors in one dedicated fusion center. Unfortunately, this introduces a single point of failure when the fusion center is not available and, therefore, can cause outages within the network. Moreover, sensor networks often feature complicated and varying network topologies without a central fusion node. Therefore, this dissertation proposes methods to keep track of correlated estimation errors independent of the underlying network

structure. A disadvantage of reconstructing cross-covariances using samples or decompositions is the communication of additional information that causes growing requirements in processing, memory and bandwidth. To limit communication requirements, this thesis proposes a sliding window approach that partially discards information about correlated noise terms. Further, it proposes methods to address the resulting partial knowledge of correlations using conservative approximations.

However, there are cases where the full reconstruction of correlated estimation errors is impossible. On the one side, this is the case when various system models are possible that result in a set of correlations, e.g., when the number of local processing steps is uncertain. On the other side, this also applies when there is no knowledge about the underlying system parameters, or when the proposed reconstruction algorithms cannot be executed due to limited computational power. In this case, a simulation-based approach can be adopted to estimate the correlation. This thesis proposes a simulation approach that can estimate the correlation between estimation errors for systems in which correlated estimation errors cannot be reconstructed. Further, it proposes an analytic approach to fully or partially reconstruct correlated estimation errors for several possible system configurations. Finally, this thesis derives methods to exploit partial knowledge, either from estimated correlations or for sets of fully or partially reconstructed correlations. Further, this thesis proposes a Gaussian mixture approach to use sets of correlations directly.

The methods proposed in this thesis offer users and a manufacturers of distributed estimation systems a wide variety of systematic approaches to deal with unknown correlated estimation errors from various sources and with different levels of uncertainty. Depending on the kind of partial knowledge, the communication requirements, and the needed quality of the fusion result, practitioners in distributed estimation can pick a method that perfectly fits their application. Hence, this thesis fills a pressing gap in the literature that has great potential to open new applications for distributed sensor systems.

CHAPTER 1

Introduction

Contents

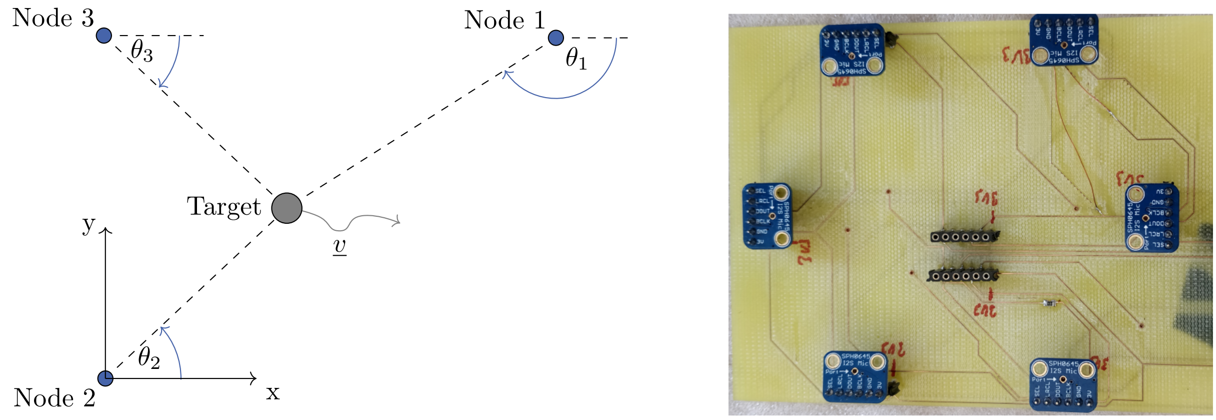
1.1 Context 1
1.2 Problem Formulation 3
1.3 Related Work 4
1.4 Open Challenges of Data Fusion in Sensor Networks 5
1.5 Contributions and Outline 6

Many sensor networks have emerged from military research and development [33, 39]. While military applications are still a key aspect, sensor networks are also becoming a vital technology in many other applications, e.g., environmental and habitat monitoring, traffic control [61] or maritime surveillance to control marine transport or to combat smuggling activities [21]. Furthermore, sensor networks are also present in many industrial applications [41] for the monitoring of air, water, waste, the condition of buildings and machines or to control industrial processes and resources.

Because of the wide variety of different tasks and application fields, sensor networks are facing various challenges. In many applications it is expected that sensor networks execute their tasks autonomously without intervention of a human operator. This means, that they need to work self-sufficiently, using their own energy supply, memory and take care of necessary communication themselves. Sensor networks are required to work reliably under different conditions and often in harsh environments without constant supervision. Additionally, they need to be versatile, simple to use and install, and have long life time. The last two decades have seen a growing trend towards low-cost sensor nodes [61] based on small embedded devices. These devises are often supplied by batteries or renewable energy sources, e.g., solar panels and therefore come with a limited energy supply. Furthermore, they often contain low-cost hardware with limited memory and computational power. Since the bandwidth in sensor networks is usually limited and the exchange of data uses a significant amount of energy, communication is a bottleneck for many applications [39].

1.1 Context

Sensor networks contain a number of sensors that are often spatially distributed to better monitor the phenomenon of interest. The sensors are able to capture different forms of information such as distances, bearings, temperatures, or images. Yet, the sensor network does not monitor all



(a) Test setup for the acoustic localization using three sensor nodes measuring the bearing θ towards a single target that is moving with velocity $\underline{v}$.

(b) PCB of the used microphone sensor array to determine the angle towards a sound source.

Figure 1.1: Experimental setup for acoustic bearings-only tracking (adapted from[145]).

available information, but a small subset of interesting parameters of the system, e.g., the position or velocity of a moving object.

The observed system can be characterized by the system state, which contains the information or variables about the process we are interested in. Usually, the time evolution of the system cannot be modeled deterministically [71] as there are uncertainties about the system model or the input to the system. Therefore, it is modeled as a dynamic system excited by noise. Because of this noise, the uncertainty about the system's state grows over time. Thus, additionally, information is provided by the sensors to gain knowledge about the state of the system and decrease uncertainty. However, measurements obtained by sensors are also subject to noise. Hence, estimation, or also referred to as filtering, methods, e.g., the popular Kalman filter [71] are used to estimate the state of the system and minimize the uncertainty. A special kind of estimate is called a track which often refers to the position, heading, or velocity of a moving target.

By distributing sensor nodes over a wide area, the sensor network is able to capture more information about the system, e.g., by providing different angles towards a moving target based on the position of the sensor. Sensor networks also often contain heterogeneous sensor nodes that differ in the measurement principle, the type of sensor, or the operating point. An example for a system with distributed sensor nodes is depicted in Figure 1.1. In this application, several sensor nodes are equipped with microphone arrays that each contain six microphones (see Figure 1.1b) that pick up the sounds emitted by a moving target. In addition, each sensor node is equipped with an embedded PC to allow the local storage and processing of recorded audio sequences. Furthermore, the sensor nodes can communicate over wireless technology, e.g., Zigbee or WiFi. Because of the limited communication bandwidth, it is infeasible to send audio tracks of every sensor array to a central processing unit. However, audio tracks can be processed locally to obtain measurements of the bearings towards the target [143, 145] (see Figure 1.1a).

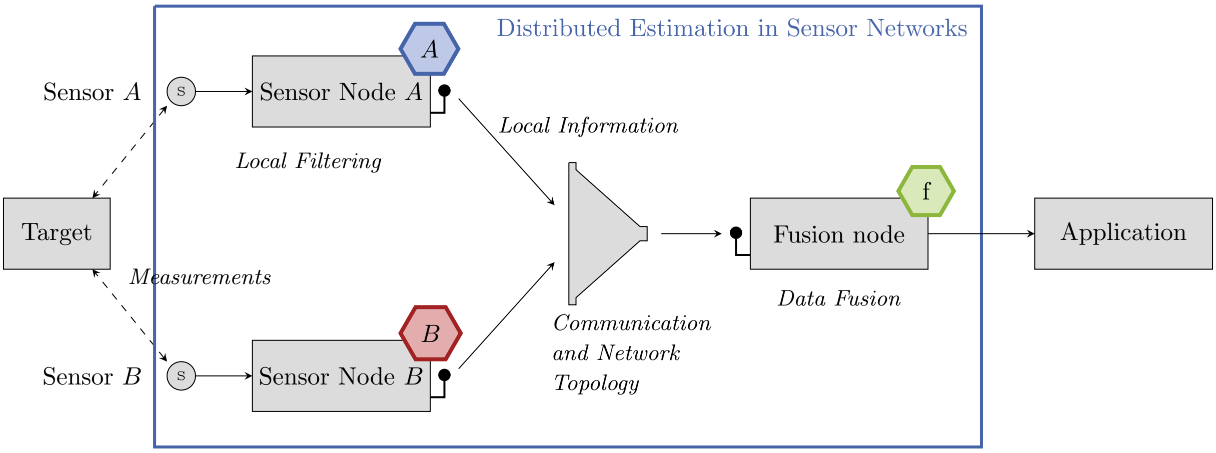


Figure 1.2: Processing steps for distributed estimation in sensor networks, two sensor nodes A (blue) and B (red) and a fusion node f (green).

1.2 Problem Formulation

Because of the spatial distribution, sensor nodes can only collect locally available data that capture a tiny amount of information about the complete system state. In the example from Figure 1.1, the locally available bearings towards the target are not sufficient to recover the position of the target. Hence, local data have to be fused to obtain the bigger picture about the observed system. There are different possibilities to design this distributed setup. For example, it is possible to collect all measurements from local sensors and send them to a central filtering node that optimally estimates the state. While this approach results in the most accurate and least uncertain estimate, it is often impractical in large sensor networks. Since measurements are taken with a high update rate, the central processing results in a significant amount of data that needs to be communicated. Furthermore, measurements can get lost or delayed due to network effects and the utilization of a single filtering node introduces a single point of failure.

An alternative approach is distributed estimation, visualized in Figure 1.2. Sensor nodes A and B do not send raw measurements to a central node. Instead, measurements are processed in local filters, recursively incorporated into state estimates that account for the uncertainty of the data and observed process. Hence, the local state estimates contain the history of previously obtained measurements. Therefore, no information is lost if the network between sensor nodes fails since the information is still present in the local sensor nodes. The state estimates are then send to a fusion node. Distributed estimation leads to more robust systems because no single point of failure exists. It is also flexible since new sensor nodes can be added and removed without reconfiguring the complete system. Lastly, distributed estimation is more scalable because the computational power of a central node does not have to scale with the size of the network. Instead, computational demands are split and distributed to the local nodes that can be designed in a modular fashion [57]. Because of limited local information, sensor nodes rely on information from other sensor nodes. On the one side, local estimators use the same assumptions about the process. On the other hand, sensor nodes also share local measurement information that is incorporated in the state estimates that are shared and fused. Therefore, the downside of the distributed estimation is common information resulting in correlated estimation errors. This double-counting of information is also known as the Track-to-Track Fusion (T2TF) problem, which was first studied in [11]. These correlated estimation errors must be accounted for during the fusion of local state estimates to obtain correct fused estimates.

Two aspects for the fusion result are important when fusing state estimates. First, the fused estimate should be **accurate**, meaning that it has to be as close as possible to the system's actual state, resulting in a minimal estimation error. Second, the estimate has to be **credible** [83]. The term consistency [10] is sometimes also used in this context. However, consistency is an already well-established concept in statistics with a different meaning. An estimator can be seen as credible at a certain level when the actual error between the true system state and the estimated state has no statistically significant difference to the estimated uncertainty, e.g., the covariance matrix. This means that the fused estimate should not overly under- or overestimate the uncertainty. Therefore, depending on the application, one can define a certain confidence level for the difference between estimation error and estimated uncertainty for which an estimator is deemed credible. An estimator is optimistic when the estimated uncertainty is smaller at a certain level than the actual estimation error, while it is conservative (or pessimistic) when it overestimates its uncertainty at a certain level.

1.3 Related Work

There is a growing body of literature that covers different aspects of sensor networks. The following section gives a brief overview of related work about sensor networks and distributed estimation. A more thorough discussion of related work is given in the individual chapters. The authors in [127] discuss several important aspects of sensor networks such as limited communication resources, data quantization, random transmission delays, packet dropouts, fading measurements, or communication disturbances. Sensor networks are also closely related to cyber-physical systems. A survey on various challenges and applications is given in [72]. A specific topic of networked data processing is target tracking, where several sensor nodes track the trajectory or the pose of one or several moving targets. These tracking problems also cover other topics such as data association or sensor management [77]. As discussed earlier, sensor networks are required to operate autonomously and therefore work reliably at all times. This requirement for reliability also includes a robust and credible fusion of local information from distributed sensor nodes.

Distributed estimation has been an object of research since the 1980s [32]. While distributed estimation improves the robustness of the sensor network, it also introduces new challenges. The authors in [58] provide a comprehensive collection of chapters about different aspects and methods of distributed estimation for network-centric operations. A survey on advances in distributed filtering and state estimation problems by [39] also touches topics such as network topology, network-induced phenomena, and power constraints that influence the filtering performance. Many existing fusion methods are based on the information form of the Kalman filter [94, 95, 98]. The information form can be used to optimally distribute Kalman filters over the network of nodes [56, 60, 76, 114, 122] and similar approaches are presented in publications about the channel filter [30, 57] or the Information Matrix Fusion (IMF) [131]. In cases where the sensor nodes are not fully connected and do not have a full update rate, they tend to be suboptimal, leading to underestimating uncertainty.

A special class of fusion algorithms for sensor networks aims to converge to a global estimate by iteratively exchanging information between neighboring nodes. Prominent examples of such algorithms are consensus on measurements [104], consensus on information [15, 105] or hybrid approaches [17, 87]. Consensus methods can be regarded as suboptimal fusion rules [31]. A similar class of suboptimal fusion rules are diffusion methods [22, 62]. While diffusion methods have similarities with consensus methods, they are better suited for highly dynamic problems because they do not wait until a consensus is reached [31].

Instead of using the information form for the fusion, it is also possible to fuse state estimates directly. In this thesis, we will mainly focus on the fusion of state estimates from distributed sensor nodes. As discussed earlier, this problem is also known as T2TF [11, 12, 29, 102]. In many applications, state estimates have correlated estimation errors due to common prior information and common process noise. However, knowledge about the correlation of estimation errors is needed to fuse tracks properly, and neglecting this fact during the fusion steps leads to underestimating the uncertainty of the fused estimate [34]. The interdependency of correlated estimation errors is usually unknown. Several methods propose ways to handle the missing knowledge. The most prominent of these algorithms is Covariance Intersection (CI) [27, 67, 111]. CI can fuse state estimates with any correlation while always yielding fusion results that are guaranteed to never underestimate the uncertainty. However, state estimation errors are usually not fully correlated since fusion is only valuable when nodes contribute new information. Hence, the performance of the fusion can be enhanced by considering a smaller set of admissible correlations. Several methods propose less conservative fusion results, such as Ellipsoidal Intersection (EI) [120], Inverse Covariance Intersection (ICI) [101, 103], or methods for improved parameterization of CI [2, 106, 113].Furthermore, other authors formulate fusion rules using optimization-based approaches [24, 139].

Many methods that use conservative approximations, e.g., CI or ICI, still obtain much more conservative fusion results, while certain assumptions about the dependency of estimation errors, e.g., EI, may underestimate the uncertainty [3, 100]. However, the fusion using the interdependency of correlated estimation errors is advantageous because it allows optimal fusion of state estimates. In recent years, several methods for the reconstruction of cross-covariances using ensembles [37], deterministic samples [125], or decompositions [110] have been proposed. These methods require additional information to reconstruct correlated estimation errors, leading to a trade-off between optimality and network capacity.

1.4 Open Challenges of Data Fusion in Sensor Networks

The fusion of state estimates from distributed local sensor nodes is still a challenge in many sensor networks. One of the main issues is the correct handling of **correlated estimation errors**, which is vital for a credible and robust fusion. Neglecting these correlations results in an overconfident fusion result where the uncertainty of the state estimate is assumed to be smaller than the actual error. In the worst case, this can lead to filter divergence and bring down the complete sensor network. In linear systems, it is possible to reconstruct the current correlation between estimation errors analytically by using knowledge about the processing steps and the parameters of the local filters. However, sensor networks often contain different sensor systems, and sensor nodes differ in their local computation. Moreover, local state estimates might relate to each other by known transformations or share different amounts of information. Therefore, the information present in local nodes can be **heterogeneous** [136, 137, 138], which has to be accounted for when calculating correlated estimation errors.

Distributed estimation requires information to be shared between sensor nodes, e.g., by using a wireless network. The flow of information between sensor nodes is highly dependent on the utilized **network topology**. For example, data may be collected in a central node or passed on through several nodes to finally reach the hierarchically highest nodes. There might be no predefined hierarchy in other cases, and sensor nodes collect information only through their neighboring nodes. While information flow is easier to keep track of in networks with a central or hierarchical structure, the aim in many sensor networks is to create redundancy and robustness. Therefore, decentralized sensor networks are the best way to achieve autonomous and robust systems. Since the information flow is unregulated in such networks, information loops occur where the sensor nodes receive previously incorporated information. Thus, the correlation of estimation errors also has to be kept track of in networks with complicated topology and possibly many sensor nodes.

As discussed previously, sensor networks often consist of low-cost sensor nodes with limited energy supply. Another limiting factor is the **communication bandwidth**. Since communication consumes a significant amount of energy, information passed between sensor nodes must be as valuable as possible. Furthermore, the sensor nodes cannot exchange information at all times and between all sensor nodes simultaneously. Therefore, fusion algorithms have to be robust to these limitations. Consequently, local sensor nodes have to reduce the amount of data to make communication as efficient as possible.

The subsequent reduction on information needs to be done in a fashion that does not negatively affect the credibility and robustness of the fused estimate. Since the full knowledge about correlated estimation errors is not available when the reconstruction is incomplete, fusion methods need to account for the missing information to ensure credible fusion results. Therefore, only **partial knowledge** about the correlation of estimation errors is available for the fusion. Partial knowledge can be represented in different forms [4]. The estimate can be split into two parts, representing known or unknown correlation, as proposed by Split Covariance Intersection (SCI), or a set of admissible correlations can be utilized [59, 108]. Other approaches use heuristics, e.g., EI, or implicit bounds on the common information, e.g., ICI. The calculation of these sets of admissible correlation coefficients is still an open research question but bears great potential for improving the fusion and obtaining tight bounds.

1.5 Contributions and Outline

The contribution of this thesis is a systematic framework for the fusion of state estimates from distributed sensor nodes that accounts for correlated estimation errors in the form of cross-covariances or correlation coefficients. In this thesis, the term distributed means that the global computational effort is distributed to several instances that execute local computations with limited information. As discussed, this is the case for sensor nodes that are spatially distributed within a sensor network. However, a distributed system can also contain nodes at the same location, but where the computation is distributed to several entities, e.g., for parallel computation on different processor cores. While parallelization is also applicable to the proposed solutions, this thesis mainly refers to spatially distributed sensor nodes. The local processing leads to correlated estimation errors that are usually unknown. Therefore, this thesis proposes several essential contributions. First, methods for fully reconstructing correlated estimation errors using information inferred from the local state estimates are proposed. These approaches are extended to reconstruct correlated estimation errors in systems with different state-space representations and network topologies. Then, the partial reconstruction of correlated estimation errors is proposed to alleviate bandwidth considerations. Furthermore, approaches to learning sets of correlated estimation errors using an analytic and a simulation-based approach to obtain partial knowledge about correlated estimation errors are developed. Last, approaches to utilize this partial knowledge to obtain accurate and credible fusion results are proposed. In this thesis, discrete-time systems and a linear fusion framework are considered. The thesis is structured as follows.

Chapter 2 - Distributed Estimation: gives a brief introduction to the problem of distributed state estimation. First, the basics of state estimation and communication in sensor networks relevant to the fusion step are provided. Afterward, the linear fusion of state estimates is introduced, and sources for correlated estimation errors are discussed. This chapter further discusses reasons and types of uncertainty about correlated estimation errors in distributed estimation. Last, different strategies and open challenges to address these uncertainties and gain more knowledge about the correlated estimation errors are given, motivating the rest of this thesis.

Chapter 3 – Reconstruction of Cross-Covariances: introduces methods to analytically calculate correlated estimation errors in the form of cross-covariance matrices that are necessary to fuse state estimates optimally. First, the chapter focuses on the reconstruction of cross-covariance matrices for systems with a centralized fusion center. Then, based on this initial discussion, two methods are introduced and compared that optimally keep track of cross-covariances in a distributed fashion. The first method is based on square root decomposition of correlated noise terms, while the second method uses deterministic samples. Afterward, both methods are extended to the fusion of state estimates and the track-keeping of cross-covariances in distributed estimation tasks with heterogeneous local state representations that are subject to linear transformations. Last, the methods are generalized for the track-keeping of cross-covariances for fusion in hierarchical and fully decentralized sensor networks.

Chapter 4 – Partial Reconstruction of Cross-Covariances: continues the reconstruction of cross-covariances. Both distributed track-keeping methods for cross-covariance from the previous chapter require the communication of a significant amount of information that grows over time. This increase in data can be problematic in many sensor networks as they suffer from limited communication bandwidth and energy. Therefore, this chapter proposes methods to reduce communication requirements by limiting the amount of tracked information. Furthermore, since the discarded information is only partially available to the fusion center, methods are proposed that account for this partial knowledge about correlated estimation errors. The proposed framework leads to a trade-off between communication bandwidth and the quality of the fusion result. This approach is also extended to the fusion in hierarchical and fully decentralized sensor networks.

Chapter 5 – Learning Partial Knowledge: discusses systems where the complete or partial reconstruction of cross-covariances is not possible. For example, this happens when sensor systems have been bought from a third party distributor and therefore do not allow a later implementation of reconstruction methods. However, when the behavior of the sensor system is known, it might be possible to simulate the state estimation and fusion step and estimate the correlation between state estimates. First, this chapter discusses the natural constraints of correlation coefficients that depend on each other to form a valid joint distribution. Afterward, approaches to estimate the correlation from noisy observations of the state estimates are discussed and compared.

Chapter 6 – Exploiting Partial Knowledge: proposes methods to utilize learned partial knowledge. The chapter investigates two different approaches. First, sets of correlation coefficients obtained by full or partial reconstruction using systems with different parameterizations are considered. Second, systems where the correlation is estimated, as discussed in the previous chapter, are investigated. Several conservative methods that can use partial knowledge to improve the fusion result are examined. Furthermore, a Gaussian mixture approach is proposed to use the estimated correlation during the fusion step directly. Moreover, to retrieve partial information, the necessary system design and identification approaches are presented. Finally, the analytic and the simulation approach are evaluated using simple numerical examples.

CHAPTER
2

Distributed Estimation

Contents

2.1 Distributed State Estimation in Sensor Networks 9
2.1.1 Local Estimation 10
2.1.2 Communication between Sensor Nodes 11
2.1.3 Fusion of State Estimates 13
2.2 Correlation Uncertainty in Distributed Estimation 17
2.2.1 Reasons and Types of Correlation Uncertainty 18
2.2.2 Approaches for Correlation Retrieval 19
2.3 Conclusions to Distributed Estimation 20

As discussed in Chapter 1, sensor networks often contain many distributed sensor nodes. The centralized processing of measurement data is often infeasible as it requires constant communications and high computational power of the central estimator. Therefore, the processing is distributed to local estimators, exchanging local information with other nodes and fusing it to improve local estimation. The process of distributed estimation can be divided into three phases. First, locally available measurements are processed and combined with knowledge about the system behavior. Afterward, local information is exchanged with the other nodes given a specific network topology. Last, the information is gathered at a fusion center, or one of the sensor nodes, and fused. This thesis focuses on the fusion of state estimates. However, the fusion of state estimates is only possible if the correlation between estimation errors is known. Yet, there are many reasons why correlations are either unknown or uncertain. This chapter lays out the theoretical foundation for distributed estimation. Afterward, reasons and types of correlation uncertainty of estimation errors are discussed. From this discussion, approaches to retrieve full or partial knowledge about correlated estimation errors are derived, which motivates the rest of this dissertation.

2.1 Distributed State Estimation in Sensor Networks

This first section is concerned with the local estimation of a dynamic system's state, where measurements from the sensors are combined with a predicted estimate. Since one local node only has a limited amount of information, data of several sensor nodes are necessary to obtain a complete picture of the underlying process. Therefore, the local state estimates need to be sent to one or several nodes. Furthermore, the utilized network topology plays an essential role in the data flow in the sensor network. Hence, different network topologies are introduced, and the resulting implications for the data flow and the fusion are discussed. Last, this section introduces the optimal fusion of state estimates and discusses sources of correlated estimation errors.

2.1.1 Local Estimation

It is assumed that the state $\underline{\mathbf{x}}_k \in \mathbb{R}^{n_x}$ of state dimension n_x of a linear discrete-time and time-variant stochastic dynamic system evolves over time by

$$\underline{\mathbf{x}}_k = \mathbf{A}_k \, \underline{\mathbf{x}}_{k-1} + \underline{\mathbf{w}}_k \quad \text{with} \quad \underline{\mathbf{w}}_k \sim \mathcal{N}(\underline{0}, \mathbf{Q}_k), \tag{2.1}$$

where k is the time index and $\mathbf{A}_k$ is the state transition matrix. The system is subject to zero-mean white Gaussian system noise $\underline{\mathbf{w}}_k$ with noise dimension $n_w = n_x$ and covariance matrix $\mathbf{Q}_k$. Many real-world applications are also controlled by a system input $\underline{u}_k$ and an input matrix $\mathbf{B}_k$ according to $\underline{\mathbf{x}}_k = \mathbf{A}_k \, \underline{\mathbf{x}}_{k-1} + \mathbf{B}_k \, \underline{u}_k + \underline{\mathbf{w}}_k$, which is neglected here for simplicity.

The system state is observed by a network of L sensor nodes, where each node i employs a linear observation model $\mathbf{C}_k^i$ using

$$\underline{z}_k^i = \mathbf{C}_k^i \, \underline{\mathbf{x}}_k + \underline{\mathbf{v}}_k^i \quad \text{with} \quad \underline{\mathbf{v}}_k^i \sim \mathcal{N}(\underline{0}, \mathbf{R}_k^i) \, . \tag{2.2}$$

These measurements are affected by zero-mean white Gaussian system noise $\underline{\mathbf{v}}_k^i$ with measurement noise covariance matrix $\mathbf{R}_k^i$. It is assumed that measurement noises are mutually independent and uncorrelated with the process noise. Each local sensor node i processes a state estimate $\hat{\underline{x}}^i$ and a covariance matrix $\mathbf{P}^i = \mathrm{E}[(\hat{\underline{x}}^i - \underline{\mathbf{x}})(\hat{\underline{x}}^i - \underline{\mathbf{x}})^\mathrm{T}]$. A popular way to process the uncertain knowledge about the system model, prior knowledge about the state, and noisy measurements into a state estimate is the well-known Kalman filter [71]. The Kalman filter is widely used as it is the Best Linear Unbiased Estimator (BLUE) and minimizes the mean squared estimation error. The processing of the state estimate is divided into two parts. First, an a priori state estimate is predicted using the system model (2.1)

$$\hat{\underline{x}}_{k|k-1}^i = \mathbf{A}_k \hat{\underline{x}}_{k-1|k-1}$$

and the predicted covariance is

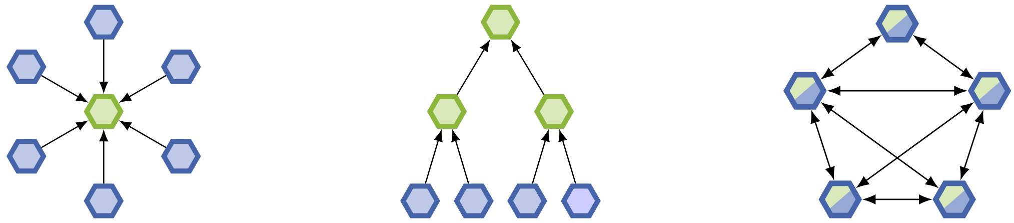
$$\begin{aligned} \mathbf{P}_{k|k-1}^i &= \mathrm{E}\big[(\hat{\underline{x}}_{k|k-1}^i - \underline{\mathbf{x}}_k)(\hat{\underline{x}}_{k|k-1}^i - \underline{\mathbf{x}}_k)^\mathrm{T}\big] \\ &= \mathbf{A}_k \mathbf{P}_{k-1|k-1}^i \mathbf{A}_k^\mathrm{T} + \mathbf{Q}_k \, . \end{aligned} \tag{2.3}$$

The predicted state estimate is then fused with the measurement $\underline{z}_k^i$ obtained in node i during the filtering step. This entails a linear combination of the predicted estimate and the innovation according to

$$\begin{aligned} \hat{\underline{x}}_{k|k}^i &= \hat{\underline{x}}_{k|k-1}^i + \mathbf{K}_k^i(\underline{z}_k^i - \mathbf{C}_k^i \hat{\underline{x}}_{k|k-1}^i) \\ &= (\mathbf{I} - \mathbf{K}_k^i \mathbf{C}_k^i) \hat{\underline{x}}_{k|k-1}^i + \mathbf{K}_k^i \underline{z}_k^i \, . \end{aligned}$$

Furthermore, the covariance after the measurement update is calculated by

$$\mathbf{P}_{k|k}^i = (\mathbf{I} - \mathbf{K}_k^i \mathbf{C}_k^i) \mathbf{P}_{k|k-1}^i (\mathbf{I} - \mathbf{K}_k^i \mathbf{C}_k^i)^\mathrm{T} + \mathbf{K}_k^i \mathbf{R}_k^i (\mathbf{K}_k^i)^\mathrm{T} \, . \tag{2.4}$$



(a) Centralized **(b)** Hierarchical **(c)** Decentralized

Figure 2.1: Different network topologies with sensor nodes (blue), nodes only dedicated to fusion (green) and sensor nodes with fusion capabilities (blue and green).

The Kalman gain $\mathbf{K}_k^i$ is obtained by

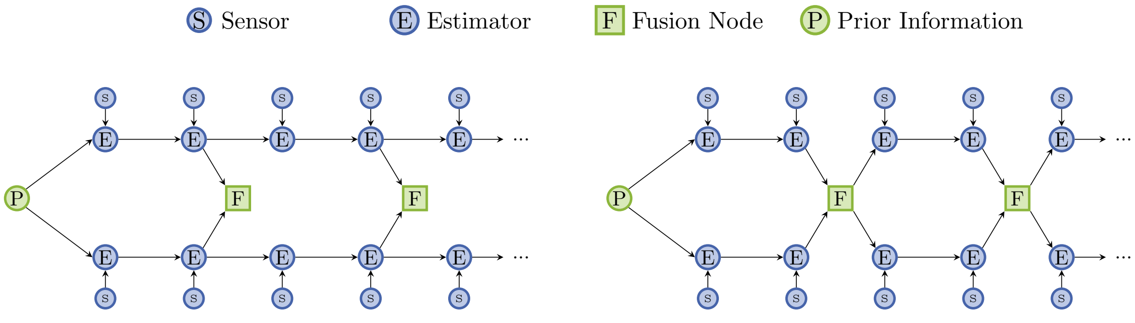
$$\mathbf{K}_k^i = \mathbf{P}_{k|k-1}^i (\mathbf{C}_k^i)^{\mathrm{T}} \left(\mathbf{C}_k^i \mathbf{P}_{k|k-1}^i (\mathbf{C}_k^i)^{\mathrm{T}} + \mathbf{R}_k^i\right)^{-1}, \tag{2.5}$$

minimizing the covariance matrix $\mathbf{P}_{k|k}^i$ and providing an optimal combination. If several measurements are available during a single time step, they can be incorporated sequentially by repeating the filtering step for every measurement, or they can be incorporated block-wise [13, 102].

2.1.2 Communication between Sensor Nodes

In general, the cooperation of sensor nodes is beneficial for obtaining the bigger picture about the underlying process and leads to more accurate and robust estimation results than a single node could provide. Therefore, the communication between sensor nodes is essential for fulfilling the objective of the sensor network. Moreover, the cooperation between nodes can benefit the estimation of a single sensor node by providing supplementary information. For example, nodes can only process a small subset of information, e.g., only angle but no distance information towards a moving target. Then, the state is not fully observable, and therefore the local estimate diverges. Knowledge from other sensor nodes can stabilize the local estimate and enable successful target tracking. Furthermore, sensor nodes can lose the line of sight towards the target, e.g., when an obstacle obstructs it. In conclusion, the communication of sensor nodes is an essential aspect of sensor networks. Therefore, communication has to be robust. The path of the information flow depends on the application, and it heavily influences the degree of correlation between estimation errors. For the sake of clarity, it is assumed that the processing steps of the estimators are perfectly synchronized and that there are no delays. It should be noted that these assumptions may be violated in real-world applications and several authors have concerned themselves with addressing the arising issues [55, 76, 135].

The following section discusses the information flow in different network topologies, namely centralized fusion, hierarchical fusion, and decentralized fusion, which can be seen in Figure 2.1. For the interested reader, the authors in [86, 109, 129] provide a detailed discussion about different network architectures and algorithms of fusion in sensor networks. A helpful tool for the analysis of the information flow in sensor networks is the information graph [35, 36]. In this section, the illustration of the information graph slightly deviates from the existing body of literature because an additional node for the prior information is introduced. Furthermore, no additional nodes for communication transmission and reception are present. Therefore, it is assumed that



(a) Centralized fusion without feedback. **(b)** Centralized fusion with feedback.

Figure 2.2: Information graph for different fusion topologies, with and without feedback from the fusion center.

estimators and fusion nodes can transmit and receive information. Moreover, feedback from the fusion node back to the estimators is assumed in some cases, which is discussed in this section. Arrows indicate the direction of the information flow, and the time index k increases from left to right. The information graph includes a node for the sensors that generate measurements. The local measurements are then processed by the local estimators as discussed in Section 2.1.1. The estimators are initialized with prior knowledge in the beginning. This information can be the same or different for all estimators. After the estimation is complete, the fusion node receives information from the estimators and fuses it. Some of the basic ideas about different fusion topologies are explained in the next section to give a short introduction that can be referred to later.

Centralized Fusion: is given when the sensor networks contain several sensor nodes that are communicating their estimates to a dedicated fusion node where they are fused, as can be seen in Figure 2.2. This is the simplest network topology for distributed estimation. Local estimators might be initialized by common prior knowledge. Then, every estimator incorporates locally available measurements into its state estimate. When the fusion step occurs, all local estimators send their estimates to the fusion node. The fusion node collects and fuses the received information. Afterward, the local estimator may carry on processing their state estimate (see Figure 2.2a), or they might receive the result of the fusion step as feedback that is used to reinitialize the local estimator (see Figure 2.2b). This feedback can also be called T2TF with memory. When the local estimates are overwritten with the fusion result, the estimation errors are fully correlated. The fusion with and without memory or feedback can have an interesting influence on the fusion result. This is investigated in [129] for the optimal fusion and in [5] for the fusion with CI.

Hierarchical Fusion: does not seek to fuse all state estimates at a single dedicated fusion center, but instead, several intermittent fusion steps are executed at several hops until a central fusion node is reached. For example, nodes might be scattered over large distances in some sensor networks. Therefore, communication with a central fusion node can be impossible if the nodes have a limited communication range. A centralized structure of the sensor networks also does not scale well when many sensor nodes are present since one node has to process information from many sensor nodes. Therefore, it can be beneficial to calculate intermediate fusion results that are passed on to the next hierarchical layer, as can be seen in Figure 2.3. When the last fusion step is executed, the fused state estimates can be passed back through the layers to the individual estimators as feedback. When the fusion step is executed with full knowledge about the correlated

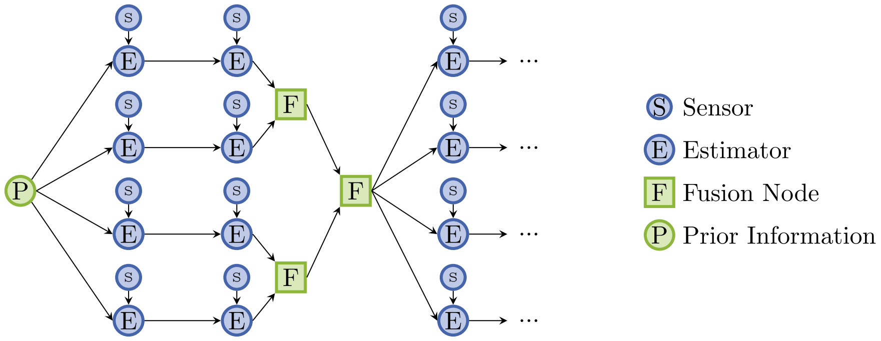


Figure 2.3: Information graph for hierarchical fusion with feedback.

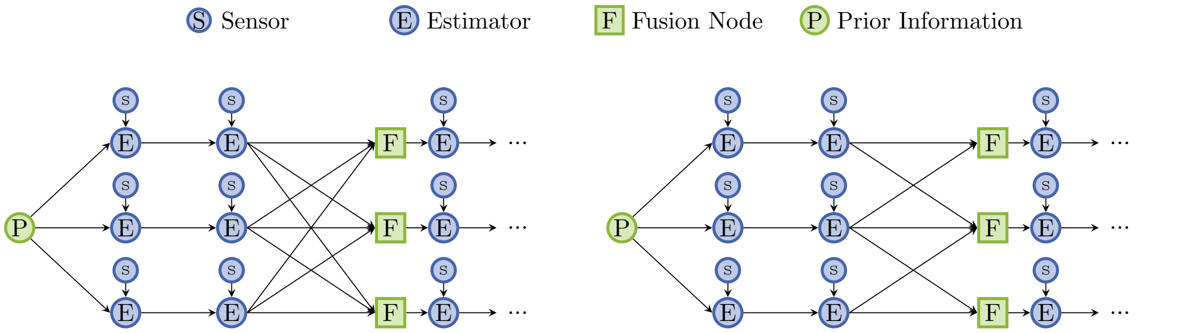
estimation errors, the fusion result is optimal and identical to the fused estimate of the centralized fusion.

Decentralized Fusion: is executed when sensor networks are employed without any hierarchy or dedicated fusion center. As before, local sensor nodes execute their respective estimation tasks. Often, these nodes work autonomously but need additional support from neighboring nodes to increase accuracy and robustness, while the information from distant sensor nodes is less valuable for the local estimation. In this case, the centralized or hierarchical network structure with one or several dedicated fusion centers can be too rigid and demand too much communication overhead. Every sensor node can serve as a fusion center in decentralized fusion networks, collect information, and fuse state estimates, as can be seen in Figure 2.4. On the downside, this implies that information from one sensor node can travel through the network and be reintroduced as new information from other sensor nodes. These information loops can cause double-counting of information that is very hard to trace back. When the network is fully connected (see Figure 2.4a), local fusion steps yield the same fusion result as the centralized fusion of all state estimates because all nodes have access to the complete information from other nodes. However, sensor networks are usually not fully connected (see Figure 2.4b). Therefore, every sensor node has different local information, which leads to partially correlated estimation errors.

2.1.3 Fusion of State Estimates

In the following section, the fusion of state estimates is discussed. For now, the problem is confined to the fusion of only two state estimates that have been received from two sensor nodes i and j. The fusion step can take place at an arbitrary time step k. In the following, the time index is omitted for convenience. The fusion can be seen as a linear combination of the two state estimates $\underline{\hat{x}}^i$ and $\underline{\hat{x}}^j$ using the fusion gains $\mathbf{F}^i$ and $\mathbf{F}^j$. Thus, the fused estimate $\underline{\hat{x}}^{\mathrm{f}}$ is

$$\underline{\hat{x}}^{\mathrm{f}} = \mathbf{F}^i \underline{\hat{x}}^i + \mathbf{F}^j \underline{\hat{x}}^j \,, \tag{2.6}$$



(a) Decentralized fusion without feedback.

(b) Decentralized fusion with only sparsely connected sensor nodes.

Figure 2.4: Information graph for decentralized fusion topologies that are fully or only sparsely connected.

which is an unbiased fusion result when $\mathbf{F}^i + \mathbf{F}^j = \mathbf{I}$. Furthermore, the error covariance matrix $\mathbf{P}^{\mathrm{f}}$ is calculated by

$$\begin{aligned}\mathbf{P}^{\mathrm{f}} &= \mathrm{E}\big[(\underline{\hat{x}}^{\mathrm{f}} - \underline{\mathbf{x}})(\underline{\hat{x}}^{\mathrm{f}} - \underline{\mathbf{x}})^{\mathrm{T}}\big] \\ &= \mathbf{F}^i\mathbf{P}^i(\mathbf{F}^i)^{\mathrm{T}} + \mathbf{F}^i\mathbf{P}^{i,j}(\mathbf{F}^j)^{\mathrm{T}} + \mathbf{F}^j\mathbf{P}^{j,i}(\mathbf{F}^i)^{\mathrm{T}} + \mathbf{F}^j\mathbf{P}^j(\mathbf{F}^j)^{\mathrm{T}} \\ &= \begin{bmatrix}\mathbf{F}^i & \mathbf{F}^j\end{bmatrix}\mathbf{J}\begin{bmatrix}\mathbf{F}^i & \mathbf{F}^j\end{bmatrix}^{\mathrm{T}} .\end{aligned} \tag{2.7}$$

The matrix $\mathbf{J}$ denotes the joint covariance matrix, which can be written as

$$\mathbf{J} = \begin{bmatrix}\mathbf{P}^i & \mathbf{P}^{i,j} \\ \mathbf{P}^{j,i} & \mathbf{P}^j\end{bmatrix} . \tag{2.8}$$

The matrices $\mathbf{P}^i$ and $\mathbf{P}^j$ stand for the covariances from the local estimators of nodes i and j. The cross-covariance matrix $\mathbf{P}^{i,j} = \mathrm{E}\big[(\underline{\hat{x}}^i - \underline{\mathbf{x}})(\underline{\hat{x}}^j - \underline{\mathbf{x}})^{\mathrm{T}}\big]$ characterizes the correlation between the estimation errors of state estimates $\underline{\hat{x}}^i$ and $\underline{\hat{x}}^j$. The fusion gains $\mathbf{F}^i$ and $\mathbf{F}^j$ are usually chosen to minimize the fused covariance matrix $\mathbf{P}^{\mathrm{f}} = \mathrm{E}\big[(\underline{\hat{x}}^{\mathrm{f}} - \underline{\mathbf{x}})(\underline{\hat{x}}^{\mathrm{f}} - \underline{\mathbf{x}})^{\mathrm{T}}\big]$. In this case, the fused state estimate $\underline{\hat{x}}^{\mathrm{f}}$ is referred to as the optimal fusion result. As discussed in [82], the fusion result can be formulated as the Weighted Least Squares (WLS) estimate

$$\underline{\hat{x}}^{\mathrm{WLS}} = \arg\min_{\underline{x}} \big[\underline{\hat{m}} - \mathbf{H}\underline{x}\big]^{\mathrm{T}}\mathbf{J}^{-1}\big[\underline{\hat{m}} - \mathbf{H}\underline{x}\big] , \tag{2.9}$$

where $\underline{\hat{m}} = \begin{bmatrix}(\underline{\hat{x}}^i)^{\mathrm{T}} & (\underline{\hat{x}}^j)^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$ is the joint state estimate. Finally, the matrix $\mathbf{H} = \begin{bmatrix}\mathbf{I} & \mathbf{I}\end{bmatrix}^{\mathrm{T}}$ determines how the local states map into the global state. The solution to the WLS problem in (2.9) is the gain matrix

$$\mathbf{F} = \begin{bmatrix}\mathbf{F}^i & \mathbf{F}^j\end{bmatrix} = (\mathbf{H}^{\mathrm{T}}\mathbf{J}^{-1}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{J}^{-1} . \tag{2.10}$$

The calculation of the fused covariance matrix is thus given by

$$\mathbf{P}^{\mathrm{f}} = (\mathbf{H}^{\mathrm{T}}\mathbf{J}^{-1}\mathbf{H})^{-1} . \tag{2.11}$$

Finally, the fused estimate is calculated by

$$\underline{\hat{x}}^{\mathrm{f}} = \mathbf{F}\underline{\hat{m}} = \mathbf{P}^{\mathrm{f}}\mathbf{H}^{\mathrm{T}}\mathbf{J}^{-1}\underline{\hat{m}} . \tag{2.12}$$

This fusion algorithm can also be generalized to the fusion of several state estimates [118]. The resulting joint covariance matrix and joint state vector for a number of L sensor nodes is constructed by

$$\mathbf{J} = \begin{bmatrix} \mathbf{P}^1 & \mathbf{P}^{1,2} & \dots & \mathbf{P}^{1,L} \\ \mathbf{P}^{2,1} & \mathbf{P}^2 & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}^{L,1} & \dots & \dots & \mathbf{P}^L \end{bmatrix} \quad , \quad \hat{\underline{m}} = \begin{bmatrix} \hat{\underline{x}}^1 \\ \hat{\underline{x}}^2 \\ \vdots \\ \hat{\underline{x}}^L \end{bmatrix} .$$

Furthermore, the mapping matrix is given by $\mathbf{H} = \left[\mathbf{I}, \mathbf{I}, \dots, \mathbf{I}\right]^{\mathrm{T}}$. The calculation of the fused estimates is then identical to the formulas given by (2.11) and (2.12).

Correlated Estimation Errors: are a challenging aspect for the fusion of state estimates in distributed estimation. The given algorithm for the fusion of state estimates can only be solved if the joint covariance matrix $\mathbf{J}$ is known. As discussed, the entries on the diagonal of the joint covariance matrices stem from the local estimators and are known. However, the entries on the off-diagonal are the cross-covariances describing how the estimation errors of $\hat{\underline{x}}^i$ and $\hat{\underline{x}}^j$ are correlated. As will be discussed in 2.2, these cross-covariances are usually unknown and depend on different factors such as the initialization, the local processing, or the communication scheme. Different sources of correlated estimation errors can be identified [102]:

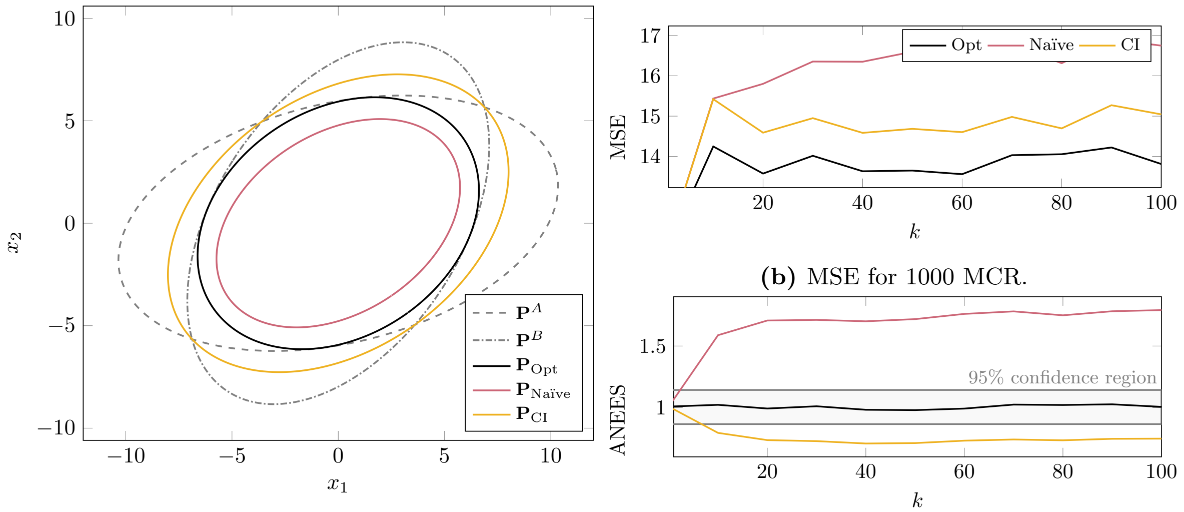
- common prior information,
- common process noise, and
- common measurement information.

Common prior information occurs when the local estimators are initialized with the same knowledge about the system, such as the same state and the same covariance matrix. Furthermore, the underlying process observed by the local sensor nodes is affected by process noise. Therefore, the local estimators incorporate the same common process noise into the prediction step. Hence, even independently initialized nodes have correlated estimation errors when estimating the same system state.

During the filtering step, local measurements are recursively incorporated into the state estimates by linear combination. Furthermore, the individual measurements can be incorporated into several other state estimates when local information is communicated through the network. If these state estimates from other nodes are passed on, information can be looped back to its source, thus resulting in double-counting of sensor data that causes additional correlations. The sources of correlated estimation errors, therefore, also depend on the network topology. For example, centralized and hierarchical fusion topologies usually contain correlations due to prior information and common process noise, while decentralized sensor networks also contain information loops that lead to double counting of measurement information.

Consequences of Correlated Estimation Errors for the Fusion Result: To demonstrate the problem of correlated estimation errors, a short example is given as follows.

Example 1: Optimal and Suboptimal Fusion of State Estimates *A system with two sensor nodes A and B is considered. The sensor nodes estimate the state of a linear time-invariant system*



(b) MSE for 1000 MCR.

(a) Comparison of error ellipses resulting from different fusion methods.

(c) ANEES for 1000 MCR.

Figure 2.5: Comparison of different fusion methods, error ellipses and simulation.

model with $\mathbf{A} = [1, 1; 0, 1]$ *and process noise covariance* $\mathbf{Q} = \mathbf{I}$*. Both estimates are initialized with* $\mathbf{P}_0 = 5\mathbf{Q}$ *and* $\hat{\underline{x}}_0 = \underline{0}$ *at time step* $k = 0$*. Furthermore, the sensor nodes use a linear measurement model* $\mathbf{C}^A = [1, 0]$ *and* $\mathbf{C}^B = [0, 1]$ *and measurement covariance* $\mathbf{R}^A = \mathbf{R}^B = 50$*. The system performs* 10 *processing steps and then the estimates are sent to a fusion center. For the performance evaluation* 1000 *Monte Carlo Runs (MCR) are executed. After every fusion step, the fusion result is communicated back to the estimates as feedback to reinitialize the local estimates. Afterward, the local estimates continue their estimation.*

Three fusion methods are compared in this example. The first method is the naïve fusion which neglects the cross-correlation. The second is CI, which bounds all possible fused covariance matrices and is guaranteed to provide results that never underestimate the uncertainty. Last, the Bar-Shalom–Campo (BSC) formulas (Opt) are used with the actual cross-covariance matrix that was calculated analytically. The estimation error is visualized in Figure 2.5a using the error ellipse [49] of the fused covariance matrices. This short example shows that neglecting correlated estimation errors leads to a covariance matrix that is smaller than the fused covariance matrix using the actual correlation between estimation errors. Therefore, this naïve fusion approach results in underestimation of the uncertainty. The CI approach on the other hand bounds all possible fused covariance matrices. Yet, it is significantly larger than the optimal fusion result and therefore overly conservative.

The performance of the fusion also can be evaluated using M MCR of the system. In this example, the local estimators are reinitialized with the result of the fusion step (see Section 2.1.2 and Figure 2.2b). Figure 2.5b shows the Mean Squared Error (MSE) of the fusion results. The optimal fusion shows the lowest error, followed by CI and then naïve fusion. The credibility of the fused estimate is evaluated using the Averaged Normalized Estimation Error Squared (ANEES) [83], where the ANEES of time step k and simulation run ℓ is defined as

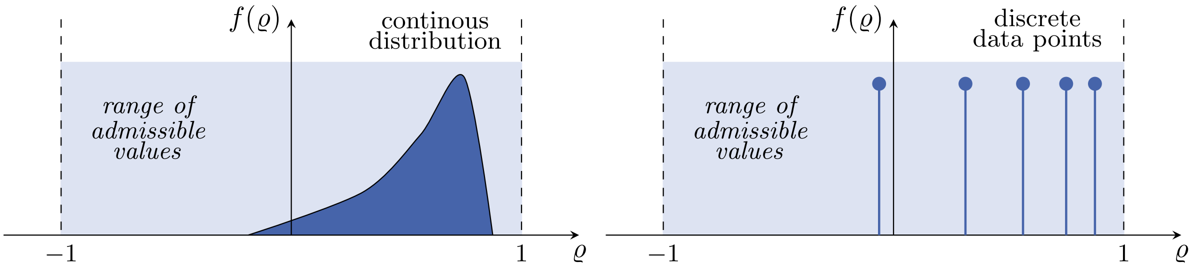
$$\text{ANEES}(k) = \frac{1}{n_x M} \sum_{\ell=1}^{M} \left(\underline{\mathbf{x}}_k(\ell) - \underline{\hat{x}}_k^{\mathrm{f}}(\ell)\right)^{\mathrm{T}} \mathbf{P}^{\mathrm{f}}(\ell)^{-1} \left(\underline{\mathbf{x}}_k(\ell) - \underline{\hat{x}}_k^{\mathrm{f}}(\ell)\right) ,$$

with the system state $\underline{\mathbf{x}}_k$, the fused estimate $\underline{\hat{x}}^{\mathrm{f}}$, and covariance matrix $\mathbf{P}^{\mathrm{f}}$ of state dimension n_x. The ANEES indicates whether the estimated uncertainty fits the actual MSE of the fused estimate and it is a measure often used in T2TF problems [14, 92]. When it is below 1, it suggests that the uncertainty is overestimated, while it is underestimated for values above 1. Furthermore, a confidence interval can be given. In this thesis, the 95% confidence interval is always plotted when the ANEES is evaluated. Figure 2.5c shows that only the optimal fusion result is within the 95% confidence interval, while CI is below (uncertainty is overestimated) and naïve fusion is above (uncertainty is underestimated). Summarizing the findings from this smaller example, using the actual correlation between estimation errors is advantageous as it yields the optimal fusion result, neither over- or underestimating the uncertainty.

Optimal Fusion of State Estimates: is a term that is used several times in this thesis. In the past, several authors have concerned themselves with the notion of optimality in distributed estimation. However, depending on the aspect, optimal fusion can have a different meaning. Whenever this thesis refers to optimal fusion, it means the optimal linear combination of state estimates in the WLS sense [82] or the Maximum Likelihood (ML) sense [23]. Nonetheless, it needs to be stressed that this is not equal to the Linear Minimum Mean Square Error (LMMSE) sense in which a Kalman filter is optimal [102]. Fusion methods based on the information form, e.g., IMF [131], can be optimal under certain assumptions in the same way a Kalman filter is optimal. This thesis distinguishes between a central Kalman filter that receives all measurements from the sensor nodes and processes them into an optimal state estimate in the LMMSE sense and the optimal fusion of state estimates or tracks.

2.2 Correlation Uncertainty in Distributed Estimation

Correct fusion of state estimates is only possible when cross-covariances between state estimates are known. The previous sections also discussed that the main reasons for correlated estimation errors are common prior information and process noise, and shared measurement information. However, even when the mechanisms behind correlated estimation errors are well established, there are different reasons why these correlations are often unknown or uncertain in many practical applications. This thesis considers the correlation of estimation errors in two forms: First, cross-covariance matrices, and second, correlation coefficients referring to the normalization of the cross-covariance matrix. Correlation coefficients are advantages because they only include admissible values in the interval of $[-1, 1]$. Furthermore, when several correlation coefficients are possible, this is referred to as sets of admissible correlation coefficients. Further explanation about the correlation and its natural constraints is given in Chapter 5. However, only cross-covariance matrices are considered to reconstruct correlated estimation errors, as they can be directly calculated from the system model and covariance matrices and used to construct the joint covariance matrix.



(a) Continuous Correlation Uncertainty.

(b) Discrete Correlation Uncertainty.

Figure 2.6: Types of Correlation Uncertainty.

The first part of this section discusses different reasons for unknown or uncertain correlated estimation errors. The reason for this lack of information determines how the missing knowledge can be fully or partially recovered. Therefore, the following section discusses different approaches to identify correlated estimation errors.

2.2.1 Reasons and Types of Correlation Uncertainty

The following section gives a systematic review of possible reasons and types of uncertainty about correlated estimation errors [8]. The resulting uncertainty is categorized into continuous and discrete correlation uncertainty.

Continuous Correlation Uncertainty: arises from uncertainties about the local estimators or the fusion steps that result in a probability distribution of the correlation rather than discrete values, as can be seen in Figure 2.6a. For example, this is the case when local sensor nodes are subject to linearization, e.g., using the Extended Kalman Filter (EKF) or the Unscented Kalman Filter (UKF). Then, the local linearization depends on the current state estimate, which affects the measurement update and the correlation between estimates. Since a fusion center or other sensor nodes do not know the local state, they also do not know the linearization point. Therefore, the uncertainty of the correlation is continuous in this case because the uncertainty of the state estimate is continuous as well.

Another reason for continuous correlation uncertainty can be model parameters that lie in a continuous but bounded range. For example, the measurement covariance matrix might depend on the current temperature, unknown to other sensor nodes. Alternatively, the exact measurement covariance matrix is not known a priori but is determined during a calibration routine. Usually, the range for the uncertain parameter can be specified, while the exact value is unknown. These model parameters might also include the system matrix, the process noise, or the prior information. Last, a reason for continuous correlation uncertainty can be due to the estimation of the correlation that is subject to noise. The uncertainty can be reduced by taking more measurements to improve the estimation process. However, in general, the resulting estimate will always be uncertain to some degree.

Discrete Correlation Uncertainty: emerges whenever correlated estimation errors can have a set of discrete value (see Figure 2.6b). As discussed for the continuous correlation uncertainty, model parameters can be uncertain or in a bounded range. Since the distribution calculation is cumbersome, one could discretize the correlated estimation errors over a suitable grid, which would result in a set of discrete values. A situation where discrete correlation uncertainty can occur naturally is when the model parameters are known, but the number of processing steps until the fusion step is executed are unknown. Because of the discrete fusion events, there are several possible correlations. Other uncertain events are the number of local sensors, the number of nodes that contributed to intermediate fusion results, the network topology, or if information or feedback has been sent successfully.

2.2.2 Approaches for Correlation Retrieval

The previous section discussed different reasons and types of correlation uncertainty. This section introduces two approaches that can be followed to obtain knowledge about correlated estimation errors in distributed estimation. Whenever the true correlated estimation errors are available or reconstructed, this is referred to as full knowledge. Furthermore, it is referred to as partial knowledge whenever the correlated estimation errors are uncertain, or when there are sets of possible correlated estimation errors. Some of the considerations of this section have been proposed in [150]. First, this section introduces an analytic approach to fully or partially calculate correlated estimation errors. Second, estimation of correlated estimation errors is discussed using a simulation-based approach.

The Analytic Approach: can be applied in cases where knowledge about the system or the set of possible system parameters can be used, e.g., varying fusion steps, prior information, system or measurement models, uncertainty about the network connection, or topology. In these cases, the cross-covariance matrix can be calculated analytically [11] for all possible combinations of system parameters. If the number of system parameters is too high, a convenient grid of system models can be used instead. This approach is applicable for many cases of continuous or discrete correlation uncertainties. However, this approach requires the calculation of the cross-covariance matrix. The calculation can be done in a centralized way, where the knowledge about the local estimators and the fusion step is used to reconstruct the cross-covariance optimally. Unfortunately, this centralized calculation requires much information and high update rates, infeasible in many sensor networks. Therefore, distributed reconstruction, e.g., using samples [125] or square-root decompositions [110], can be helpful. In some cases, not all of the knowledge necessary for the complete reconstruction is available. Therefore, cross-covariances may be only partially reconstructed, which adds additional uncertainty to the analytically calculated cross-covariances.

The Simulation Approach: can be used when the system parameters are not known. Uncertainties of the system, e.g., when nodes do not receive information from neighboring nodes, can, for example, be modeled by uncertain initial conditions. When it is possible to model the behavior of the sensor network, including the local estimation steps and the fusion, then several MCR of the complete system can be executed. By simulating the local estimation step, including the measurements, the estimation error can be calculated. Thus, the necessary partial knowledge can be estimated from calculating the correlation between the errors of the simulated state estimates. The simulation approach is feasible if the system behavior only depends on a known system input for which the output can be learned. This approach requires methods to estimate the correlation coefficients that describe the dependency between estimation errors. Since the estimation process is subject to noise, the uncertainty of the estimate has to be evaluated and accounted for during the fusion step.

2.3 Conclusions to Distributed Estimation

This chapter discussed the processing steps necessary for fusing local state estimates from distributed sensor nodes. First, locally available measurements are processed by local estimators and then communicated according to the given topology of the sensor network. Then, state estimates are fused using a WLS approach. Last, the fusion result can be communicated to other sensor nodes for subsequent fusion, e.g., in hierarchical network topologies, or used as feedback to reinitialize local estimators with improved knowledge about the observed state. However, the estimation errors of the local estimates are correlated due to common prior information, common process noise, and shared measurement information. This correlation must be accounted for to ensure credible fusion results that do not underestimate the estimate's uncertainty.

Therefore, knowledge about correlated estimation errors is the key to performing an optimal fusion of state estimates and vital for the accuracy and credibility of the fusion result. However, there are different reasons why correlated estimation errors might be partially or fully unknown, leading to continuous or discrete correlation uncertainty. Analytic or simulation-based approaches can be adopted to retrieve knowledge about the correlation of estimation errors in distributed estimation. Therefore, the following chapters investigate both analytic and simulation-based approaches to obtain full or partial knowledge about correlated estimation errors to improve fusion results.

CHAPTER
3

Full Reconstruction of Cross-Covariances

Contents

3.1 Optimal Centralized Reconstruction of Cross-covariances 22

3.2 Distributed Track-keeping of Cross-Covariances 23

3.2.1 Reconstruction Based on Square Root Decomposition 24

3.2.2 Deterministic Sample-Based Reconstruction 25

3.2.3 Comparison Between the Square Root Decomposition and the Sample-Based Reconstruction 27

3.3 Track-Keeping for Different Local Coordinate Systems 28

3.3.1 Linearly Transformed State Spaces 30

3.3.2 Overlapping State Estimates 36

3.3.3 Conclusion to Reconstruction of Cross-Covariances for Estimates from Different Coordinate Systems 40

3.4 Fully Decentralized Track-Keeping of Cross-Covariances 40

3.4.1 Optimal Reconstruction for Hierarchical Network Topologies 42

3.4.2 Optimal Reconstruction for Fully Decentralized Network Topologies 43

3.4.3 Conclusion to Fully Decentralized Track-Keeping of Cross-Covariances 45

3.5 Conclusions to Full Reconstruction of Cross-Covariances 45

The optimal fusion of state estimates requires the joint covariance matrix to be known. While the covariances of the local estimators are known to the fusing node, the cross-covariance describing the correlation between estimation errors is usually unavailable. However, when the processing steps of the local estimators are known, then the cross-covariance can be calculated analytically [11]. A significant downside is that this centralized reconstruction of cross-covariances requires either preexisting knowledge of the local estimators or constant communication of the local processing steps.

This chapter focuses on different methods for the optimal reconstruction of cross-covariances in distributed estimation. To allow a deeper insight into the reconstruction of cross-covariances, this chapter starts by studying the analytic calculation in a centralized network with only one dedicated fusion node. Then, this discussion forms the basis for deriving two distributed methods for the track-keeping of cross-covariances based on deterministic samples and the square root decomposition of correlated noise covariances. Finally, both methods are compared and further extended to sensor networks with heterogeneous state representation and different network topologies.

3.1 Optimal Centralized Reconstruction of Cross-covariances

As discussed, this section focuses on the fusion of state estimates in a single dedicated node. For simplicity, this following discussion is limited to the estimates from only two sensor nodes i and j. The state estimates are initialized with the same state $\underline{\hat{x}}_0$ and covariance matrix $\mathbf{P}_0$ at time step $k = 0$. Therefore, both estimates are fully correlated at the beginning of the estimation process due to the common prior information, resulting in the cross-covariance $\mathbf{P}_0^{i,j} = \mathbf{P}_0^{j,i} = \mathbf{P}_0$.

Next, the prior state estimates are predicted by the local Kalman filters (see (2.3)) that incorporate the process noise covariance $\mathbf{Q}_k$, resulting in the predicted cross-covariance

$$
\begin{aligned}
\mathbf{P}_{k|k-1}^{i,j} &= \mathrm{E}[(\underline{\hat{x}}_{k|k-1}^i - \underline{\mathbf{x}}_k)(\underline{\hat{x}}_{k|k-1}^j - \underline{\mathbf{x}}_k)^{\mathrm{T}}] \\
&= \mathrm{E}\Big[\big(\mathbf{A}_k\underline{\hat{x}}_{k-1|k-1}^i - (\mathbf{A}_k\underline{\mathbf{x}}_{k-1} + \underline{\mathbf{w}}_k)\big)\big(\mathbf{A}_k\underline{\hat{x}}_{k-1|k-1}^j - (\mathbf{A}_k\underline{\mathbf{x}}_{k-1} + \underline{\mathbf{w}}_k)\big)^{\mathrm{T}}\Big] \\
&= \mathbf{A}_k\mathrm{E}\Big[\big(\underline{\hat{x}}_{k-1|k-1}^i - \underline{\mathbf{x}}_{k-1}\big)\big(\underline{\hat{x}}_{k-1|k-1}^j - \underline{\mathbf{x}}_{k-1}\big)^{\mathrm{T}}\Big]\mathbf{A}_k^{\mathrm{T}} + \mathrm{E}\Big[\underline{\mathbf{w}}_k(\underline{\mathbf{w}}_k)^{\mathrm{T}}\Big] \\
&= \mathbf{A}_k\mathbf{P}_{k-1|k-1}^{i,j}\mathbf{A}_k^{\mathrm{T}} + \mathbf{Q}_k\,.
\end{aligned} \tag{3.1}
$$

Afterward, the cross-covariance is updated (see (2.4)) using the Kalman filter gain $\mathbf{K}_k$ (see (2.5))

$$
\begin{aligned}
\mathbf{P}_{k|k}^{i,j} &= \mathrm{E}[(\underline{\hat{x}}_{k|k}^i - \underline{\mathbf{x}}_k)(\underline{\hat{x}}_{k|k}^j - \underline{\mathbf{x}}_k)^{\mathrm{T}}] \\
&= \mathrm{E}\Big[\big(\underline{\hat{x}}_{k|k-1}^i + \mathbf{K}_k^i(\underline{z}_k^i - \mathbf{C}_k^i\underline{\hat{x}}_{k|k-1}^i) - \underline{\mathbf{x}}_k\big)\big(\underline{\hat{x}}_{k|k-1}^j + \mathbf{K}_k^j(\underline{z}_k^j - \mathbf{C}_k^j\underline{\hat{x}}_{k|k-1}^j) - \underline{\mathbf{x}}_k\big)^{\mathrm{T}}\Big] \\
&= \mathrm{E}\Big[\big(\underline{\hat{x}}_{k|k-1}^i + \mathbf{K}_k^i(\mathbf{C}_k^i\underline{\mathbf{x}}_k + \underline{\mathbf{v}}_k - \mathbf{C}_k^i\underline{\hat{x}}_{k|k-1}^i) - \underline{\mathbf{x}}_k\big)\big(\underline{\hat{x}}_{k|k-1}^j + \mathbf{K}_k^j(\mathbf{C}_k^j\underline{\mathbf{x}}_k + \underline{\mathbf{v}}_k - \mathbf{C}_k^j\underline{\hat{x}}_{k|k-1}^j) - \underline{\mathbf{x}}_k\big)^{\mathrm{T}}\Big] \\
&= (\mathbf{I} - \mathbf{K}_k^i\mathbf{C}_k^i)\mathrm{E}\Big[\big(\underline{\hat{x}}_{k|k-1}^i - \underline{\mathbf{x}}_k\big)\big(\underline{\hat{x}}_{k|k-1}^j - \underline{\mathbf{x}}_k\big)^{\mathrm{T}}\Big](\mathbf{I} - \mathbf{K}_k^j\mathbf{C}_k^j)^{\mathrm{T}} + \mathbf{K}_k^i\mathrm{E}\Big[\underline{\mathbf{v}}_k^i(\underline{\mathbf{v}}_k^j)^{\mathrm{T}}\Big](\mathbf{K}_k^j)^{\mathrm{T}} \\
&= \mathbf{L}_k^i\mathbf{P}_{k|k-1}^{i,j}(\mathbf{L}_k^j)^{\mathrm{T}}\,,
\end{aligned} \tag{3.2}
$$

where $\mathbf{L}_k^i = \mathbf{I} - \mathbf{K}_k^i\mathbf{C}_k^i$ and $\mathrm{E}\Big[\underline{\mathbf{v}}_k^i(\underline{\mathbf{v}}_k^j)^{\mathrm{T}}\Big] = \mathbf{0}$ because the measurement noises are mutually independent. This recursive calculation of the cross-covariance is repeated during every prediction and filtering step of the Kalman filter. The newly incorporated process noise covariances during the prediction step increase the correlation between the estimation errors, while the filtering step reduces the correlation due to the multiplication with the matrix $\mathbf{L}_k^i = \mathbf{I} - \mathbf{K}_k^i\mathbf{C}_k^i$.

This recursive calculation of the cross-covariance can also be rewritten explicitly by a sum of covariances

$$
\mathbf{P}_{k|k}^{i,j} = \mathbf{T}_{0,k}^i\mathbf{P}_0(\mathbf{T}_{0,k}^j)^{\mathrm{T}} + \sum_{\tau=1}^{k}\mathbf{T}_{\tau,k}^i\mathbf{Q}_\tau(\mathbf{T}_{\tau,k}^j)^{\mathrm{T}}\,, \tag{3.3}
$$

similarly derived in [130]. Here, τ is the processing step at which a new process noise covariance matrix $\mathbf{Q}_\tau$ is included. The matrix $\mathbf{T}_{\tau,k}$ denotes the individual matrix transformations that are a result of the local Kalman filtering steps (3.1) and (3.2)

$$
\begin{aligned}
\mathbf{T}_{0,k}^i &= \prod_{\ell=1}^{k}\mathbf{L}_\ell^i\mathbf{A}_\ell\,, \\
\mathbf{T}_{\tau,k}^i &= \Big(\prod_{\ell=\tau+1}^{k}\mathbf{L}_\ell^i\mathbf{A}_\ell\Big)\mathbf{L}_\tau^i\,.
\end{aligned}
$$

When sensor networks contain many nodes, keeping track of these correlations can be cumbersome and often infeasible as it requires full communication of all processing steps. Therefore, distributed methods to keep track of cross-covariances are required. Furthermore, sharing the calculation with the individual sensor nodes makes the track-keeping better scalable and does not burden the fusion node with a high computational load.

3.2 Distributed Track-keeping of Cross-Covariances

A considerable amount of literature has been published on distributed track keeping and reconstruction of cross-covariances in distributed estimation. The following section gives a brief review of notable publications.

The authors in [37] study the correlation of estimation errors in the context of cooperative vehicle localization. They use a generalization of the ensemble Kalman filter applied in the presence of common past information shared between vehicles. The ensemble contains random samples that represent the state estimate and the observation. The authors in [115] also propose to use random samples but use them to represent estimation errors. The authors sample common information in an identical fashion, using the same seed for the random samples in every sensor node to obtain sets of correlated and uncorrelated samples. These samples are processed locally to incorporate the local estimation step and then communicated for the fusion step to reconstruct the joint covariance matrix. Because of the random samples, the reconstructed cross-covariance asymptotically approaches the actual cross-covariance matrix with an increasing number of samples.

In [125], the authors propose a method that can optimally reconstruct the cross-covariance matrix by using deterministic instead of random samples. They use the spherical simplex sampling method [65] to create sets of correlated and uncorrelated samples that are then used to reconstruct the cross-covariance matrices. The advantage of this approach is the reduced amount of samples required for the reconstruction. Furthermore, by using deterministic samples, the cross-covariance is identical to the analytically calculated one and does not require numerous random samples that still only approximate the true cross-covariance. Finally, the authors in [110] propose a method to reconstruct cross-covariances using square root decompositions of correlated estimation errors. Like the reconstruction using deterministic samples, this reconstruction yields identical cross-covariances to the analytic calculation. Moreover, it also requires only a limited amount of communication bandwidth. This method is also appealing, as the authors also propose a bounding scheme to alleviate the growing number of correlated noise terms. For completeness, a similar work by [70] proposes a Kalman filter using a pure square root process for accounting for correlated or colored noise. Here, a square root of set matrices that contains error covariance information is also updated at time and measurement updates.

This brief review of the current state-of-the-art suggests that a deterministic reconstruction is a promising approach for optimally calculating cross-covariances. Therefore, the following sections investigate the reconstruction using the square root decompositions of correlated noise covariances and the reconstruction using deterministic samples. Furthermore, many applications contain sensor nodes with different state-space representations subject to arbitrary transformations. Therefore, an extension keeping track of correlated estimation errors for local state estimates from different state-space representations is introduced. Both approaches that are introduced in this chapter exploit the same underlying principle for the reconstruction but are limited to the reconstruction in centralized sensor networks with a dedicated fusion center. Therefore, a suitable extension for decentralized sensor networks is proposed for the introduced distributed track-keeping methods.

3.2.1 Reconstruction Based on Square Root Decomposition

First, the reconstruction using square root decompositions of correlated noise covariances is investigated. The following considerations were published in [146] and further develop the previously formulated ideas of [110]. The main idea is to decompose the correlated noise covariances incorporated during the initialization and the prediction steps into a square root, e.g., using the Cholesky decomposition. Then, every node updates and saves its history of processing steps in a matrix containing all square root decompositions of common noise covariances. During the fusion step, every node transmits its state estimate, covariance matrix, and square root matrix. The square root matrix is then used to reconstruct the cross-covariances needed to construct the joint covariance matrix. Then, this information is used to fuse the local estimates according to (2.11) and (2.12).

The recursive formula of (3.3) can be reformulated as a square root decomposition of noise covariances as

$$\begin{aligned}\mathbf{P}^{i,j}_{k|k} &= \mathbf{T}^{i}_{0,k}\sqrt{\mathbf{P}_0}(\sqrt{\mathbf{P}_0})^{\mathrm{T}}(\mathbf{T}^{j}_{0,k})^{\mathrm{T}} + \sum_{\tau=1}^{k}\mathbf{T}^{i}_{\tau,k}\sqrt{\mathbf{Q}_\tau}(\sqrt{\mathbf{Q}_\tau})^{\mathrm{T}}(\mathbf{T}^{j}_{\tau,k})^{\mathrm{T}}\\ &= \sum_{\tau=0}^{k}\mathbf{\Sigma}^{i}_{\tau,\mathbf{Q}}(\mathbf{\Sigma}^{j}_{\tau,\mathbf{Q}})^{\mathrm{T}}\,.\end{aligned}$$

Here, the square root of covariances is achieved using the Cholesky decomposition. Each sensor node stores its square root terms in the matrix

$$\mathbf{S}^{i}_{k,\mathbf{Q}} = \left[\mathbf{\Sigma}^{i}_{0,\mathbf{Q}}, \mathbf{\Sigma}^{i}_{1,\mathbf{Q}}, \dots, \mathbf{\Sigma}^{i}_{k,\mathbf{Q}}\right]$$

that includes all noise terms until the current time step k.

This calculation of the square root matrix can also be done recursively. At time step $k = 0$, it is initialized with

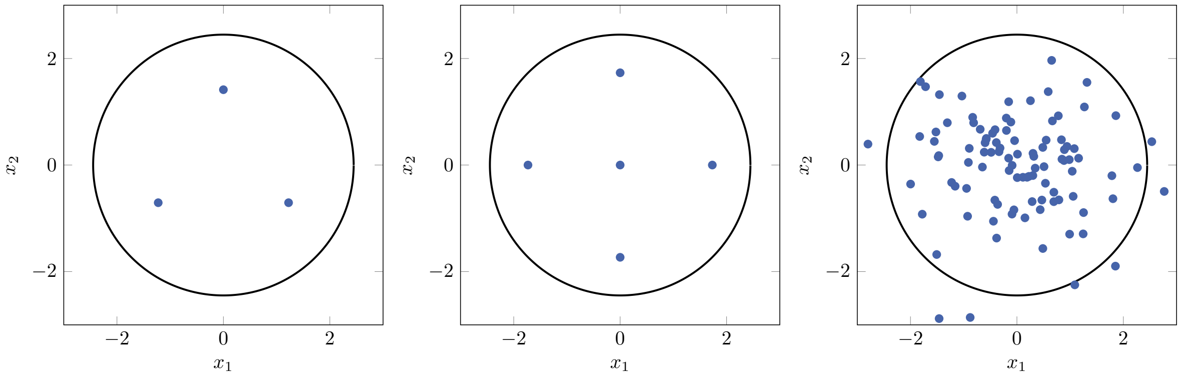
$$\mathbf{S}^{i}_{0,\mathbf{Q}} = \mathbf{\Sigma}^{i}_{k,\mathbf{Q}} = \sqrt{\mathbf{P}_0}\,.$$

Then, the matrix is propagated forward during the prediction step using the system model and a new noise term $\mathbf{\Sigma}^{i}_{k,\mathbf{Q}} = \sqrt{\mathbf{Q}}$ is included. Afterward, the square root matrix is updated using the gain matrix of the Kalman filter update $\mathbf{L}^{i}_{k} = \mathbf{I} - \mathbf{K}^{i}_{k}\mathbf{C}^{i}_{k}$

$$\mathbf{S}^{i}_{k,\mathbf{Q}} = \mathbf{L}^{i}_{k}\left[\mathbf{A}_k\mathbf{S}^{i}_{k-1,\mathbf{Q}}\ ,\ \mathbf{\Sigma}^{i}_{k,\mathbf{Q}}\right]. \tag{3.4}$$

When the fusion step is reached, the cross-covariance matrix between node i and node j is reconstructed by

$$\mathbf{P}^{i,j}_{k,Q} = \sum_{\tau=0}^{k}\mathbf{\Sigma}^{i}_{\tau,\mathbf{Q}}(\mathbf{\Sigma}^{j}_{\tau,\mathbf{Q}})^{\mathrm{T}} = \mathbf{S}^{i}_{k,\mathbf{Q}}(\mathbf{S}^{j}_{k,\mathbf{Q}})^{\mathrm{T}}\,. \tag{3.5}$$



(a) Spherical simplex sampling (3 samples)

(b) UKF sampling (5 samples)

(c) Random sampling (100 samples)

Figure 3.1: Comparison of different sampling schemes, created with the Nonlinear Estimation Toolbox [124], 95% confidence interval on the covariance (black) and samples (blue).

Hence, the calculation of the square root matrix is rather straightforward. At every prediction step, the square root matrix is predicted and a new decomposition of the process noise is included. Then, the square root is updated during the measurement step, reducing the correlation of the estimation errors. Because of the incorporation of new noise covariances, this square root matrix will grow linearly over time.

3.2.2 Deterministic Sample-Based Reconstruction

The reconstruction of cross-covariances using deterministic instead of random samples was proposed by [125]. In general, the sampling of the probability distribution is often used when nonlinear transformations are applied to the system. Therefore, the prior probability distribution is sampled and then transformed so the samples can capture the posterior distribution. The sampling scheme depends on the probability distribution and the applied transformation. Random samples (see Figure 3.1c) use randomized points of the probability distribution, which is often used in particle filters [9] where nonlinearities are severe. Deterministic samples, on the other hand, e.g., the spherical simplex sampling (see Figure 3.1a) or the UKF [68] sampling (see Figure 3.1b), use strategically chosen points that capture specific characteristics of the distribution, e.g., the mean and the covariance. These deterministic samples are suitable for reconstructing cross-covariances, as transformations by the Kalman filter are assumed to be linear, so a limited number of samples is sufficient to capture the posterior distribution correctly.

The idea behind the deterministic sampling for the reconstruction of cross-covariances is to create identical sample sets for noise covariances that are correlated in every sensor node. These sample sets are updated by the local Kalman filter steps of every sensor node. This allows the reconstruction of cross-covariance during the fusion step based on the modifications applied to the samples. The following section briefly reviews the key elements of this method using the block-wise creation of deterministic samples as it was initially proposed in [125]. Afterward, a sequential method for creating samples is discussed, where every noise covariance matrix is sampled individually.

The first step is to create an initial sample set $\{\underline{p}\}_{m=1}^{M}$ with $\underline{p}^m \in \mathbb{R}^{n_M}$ that contains a number of M samples with sample weights ω having the following characteristics

$$\sum_{m=1}^{M} \omega_m \, \underline{p}^m = \underline{0} \quad , \quad \sum_{m=1}^{M} \omega_m \, \underline{p}^m (\underline{p}^m)^{\mathrm{T}} = \mathbf{I} \,. \tag{3.6}$$

The dimension of the sample set n_M depends on the dimension of the included noise covariances and a user-defined time horizon $\mathcal{T}$ according to $n_M = n_x + (\mathcal{T} - 1)n_w$, where n_x is the dimension of state space and n_w the dimension of the process noise. The number of samples M cannot be reduced further as $n_M + 1$ to represent a valid covariance matrix in n_M dimensions [125]. The authors in [125] use the simple deterministic spherical simplex sampling method described in [65]. It is beneficial because it requires a relatively small number of samples compared to other sampling schemes. Furthermore, since samples have to be communicated to the fusion center, a small number of samples is advantageous. However, other sampling methods can also be employed as long as they share the same characteristics.

This initial sample set it then weighted with the following matrix containing the square root decomposition of prior common information and common process noise covariances

$$\mathbf{D} = \operatorname{blkdiag}\left(\sqrt{\mathbf{P}_0}, \sqrt{\mathbf{Q}_1}, \ldots, \sqrt{\mathbf{Q}_{\mathcal{T}}}\right) . \tag{3.7}$$

By weighting the sample set $\{\underline{p}\}_{m=1}^{M}$ with $\mathbf{D}$, the sample set $\{\underline{d}\}_{m=1}^{M}$ is obtained according to

$$\begin{aligned} \underline{d}_k^m &= \mathbf{D} \, \underline{p}^m \quad , \quad \forall m = 1, \ldots, M \\ &= \left[(\underline{s}_{k|k}^{i,m})^{\mathrm{T}}, \;\; (\underline{w}_{k+1}^m)^{\mathrm{T}} \;\; , \ldots , \;\; (\underline{w}_{k+\mathcal{T}}^m)^{\mathrm{T}} \right]^{\mathrm{T}} . \end{aligned} \tag{3.8}$$

This sample set includes one sample set $\{\underline{s}_{k|k}^{i,m}\}_{m=1}^{M}$ to account for the common prior information and a sample set $\{\underline{w}_{k+\tau}\}_{m=1}^{M}$ for every processing step $k + \tau$ accounting for the common process noise until a user defined time horizon $\mathcal{T}$, where $0 < \tau \leq \mathcal{T}$.

The sample sets are constructed so that multiplying samples that account for the same common information yields the underlying cross-covariance

$$\begin{aligned} \mathbf{P}_0 &= \sum_{m=1}^{M} \omega_m \, \underline{s}_{k|k}^{i,m} \big(\underline{s}_{k|k}^{i,m}\big)^{\mathrm{T}} , \\ \mathbf{Q}_{k+\tau'} &= \sum_{m=1}^{M} \omega_m \, \underline{w}_{k+\tau'}^m \big(\underline{w}_{k+\tau'}^m\big)^{\mathrm{T}} , \end{aligned}$$

while multiplying other samples that are uncorrelated to each other yields zero

$$\sum_{m=1}^{M} \omega_m \, \underline{s}_{k|k}^{i,m} \big(\underline{w}_{k+\tau'}^m\big)^{\mathrm{T}} = \sum_{m=1}^{M} \omega_m \, \underline{w}_{k+\tau'}^m \big(\underline{w}_{k+\tau''}^m\big)^{\mathrm{T}} = \mathbf{0} \,.$$

Here, τ' and τ'' are arbitrary time steps where $\tau' \neq \tau''$. The correlations between estimation errors are incorporated into the sample set by applying the transformations of the Kalman filtering steps to the sample set. During the prediction step, the sample set $\{\underline{s}_{k|k}^{i,m}\}_{m=1}^{M}$ that accounts for the common prior information is propagated forward in time. Then the sample set referring to the current process noise covariance is added, yielding

$$\underline{s}_{k|k-1}^{i,m} = \mathbf{A}_k \, \underline{s}_{k-1|k-1}^{i,m} + \underline{w}_k^m \quad , \quad \forall m = 1, \ldots, M \,. \tag{3.9}$$

The sample set is then modified by the update step of the Kalman filter gain according to

$$\underline{s}_{k|k}^{i,m} = \mathbf{L}_k^i \, \underline{s}_{k|k-1}^{i,m} \quad , \quad \forall m = 1, \dots, M \, .$$

This processing of the sample set $\{\underline{s}_{k|k}^{i,m}\}_{m=1}^{M}$ is done recursively until the time horizon $\mathcal{T}$ is reached. The cross-covariance matrix $\mathbf{P}^{i,j}$ is reconstructed by multiplying the correlation samples of node i and j

$$\mathbf{P}_k^{i,j} = \sum_{m=1}^{M} \omega_m \big(\underline{s}_{k|k}^{i,m} - \bar{\underline{s}}_{k|k}^{i}\big)\big(\underline{s}_{k|k}^{j,m} - \bar{\underline{s}}_{k|k}^{j}\big)^{\mathrm{T}} \, , \tag{3.10}$$

where $\bar{\underline{s}}$ is the mean of the sample set $\{\underline{s}_{k|k}^{i,m}\}_{m=1}^{M}$ since the processing of the samples, e.g., during the update step, can lead to a nonzero sample mean.

Instead of creating the complete sample set until a user-defined time horizon $\mathcal{T}$, it is also possible to create samples whenever needed. The samples proposed in the initial publication of [125] used weights as proposed for the spherical simplex sampling method. When more samples are needed, the weights are changed and this requires rescaling the previously created samples. This results in additional computational power, especially when many samples have already been created. We found [152] that omitting the weights is possible since the processing by the local Kalman filters is linear, and it is not critical to match the exact moments of the distribution.

3.2.3 Comparison Between the Square Root Decomposition and the Sample-Based Reconstruction

The square root decomposition and the sample-based reconstruction are both suitable approaches for tracking correlated estimation errors in distributed estimation. The underlying mechanism that results in the reconstruction of the correct cross-covariance is very similar in both methods. The construction of the deterministic samples also features the square root decomposition of correlated noise terms that are combined when multiplying the samples.

Both methods are designed to keep track of processed common information until a user-defined time horizon $\mathcal{T}$. Therefore, the sample set or the square root matrix should include common prior information and common process noise covariances of all processing steps until the next fusion step is executed. When it is assumed that the processing starts at $k = 0$ and the fusion step is $k = k^{\mathrm{f}}$, then the time horizon is $\mathcal{T} = k^{\mathrm{f}} + 1$. For the square root decomposition, the dimension of the square root matrix is $n = n_x \times (n_x + (\mathcal{T} - 1)n_w)$, where n_x denotes the dimension of the system state and n_w the dimension of the process noise. However, for sample-based reconstruction, the sampling method is important. As discussed before, the spherical simplex sampling requires less samples than other sampling methods to capture the underlying distribution and still has the required characteristics to create uncorrelated samples to other samples in the set. However, the number of samples $M = n_x \times n_M = n_x + (\mathcal{T} - 1)n_w + 1$, meaning that the dimension of the sample set is $n = n_x \times (n_x + (\mathcal{T} - 1)n_w + 1)$. Of course, other sampling methods, e.g., the UKF sampling, require more samples, meaning that the square root matrix is always smaller than the sample set.

Another important factor to consider is the usability of the methods. The square root decomposition is relatively easy to use as it only requires the computation of the Cholesky decomposition of the dependent noise term and the linear transformation of the matrix with the transformation applied by the current processing step. Moreover, the new square root decompositions are only appended to the matrix, meaning that the noise terms are in chronological order from the newest to the oldest in the matrix. Because of the update step that requires the multiplication with $\mathbf{L}_k^i = \mathbf{I} - \mathbf{K}_k^i \mathbf{C}_k^i < \mathbf{I}$, old correlated noise covariances loose their importance as they tend to approach to zero. If the noise covariances are not significant to the reconstruction anymore, they could be discarded and partially bounded, as will be discussed in Chapter 4. However, the use of deterministic samples is less straightforward in comparison. The sample set is created to have independent samples created for uncorrelated noise covariances. These samples are added to the existing set containing the current correlation of estimation errors during the prediction step. This procedure makes it cumbersome if not impossible to remove old samples when their correlated noise covariances are almost zero. Furthermore, it is not possible to observe when old samples are almost zero. A possible way to circumvent this problem is to create a sample set for every processing step individually. Thus, new samples would not be added to the old samples but appended to the sample set, similar to creating the square root matrix. Unfortunately, this would result in significantly more samples and, therefore, increase bandwidth requirements.

Summarizing these findings, we found that the square root decomposition appears to be slightly more intuitive and easier to use. Furthermore, it requires less data to be communicated, which is especially important in sensor networks, where bandwidth requirements are essential.

Track-Keeping for Nonlinear Estimators is a crucial aspect to consider. Many real-world applications do not have linear measurement models, e.g., when observing the distance or angles towards a moving target. Therefore, many applications require nonlinear filters. In order to keep track of the cross-covariance matrix, it is necessary to identify the matrix $\mathbf{L}$ that alters the predicted cross-covariance matrix during the measurement update in (3.2). This linear transformation can be found for the Kalman filter, but also all of its derivatives, e.g., the EKF or the UKF or any other regression Kalman filter [81]. The reconstruction of cross-covariances for nonlinear measurement models has been done by [125, 145] successfully for the sample-based reconstruction. Based on this investigation, [146, 148, 149] proposed a similar formulation for the square root decomposition approach, also showing credible fusion results.

3.3 Track-Keeping for Different Local Coordinate Systems

Most research on the fusion of state estimates has been carried out for estimates with identical state-space representation. However, some fusion problems do not apply to these assumptions. With an increasing number of sensor nodes spatially distributed over a large area, the computation of the complete state space can be challenging or even infeasible. However, in other cases, the local estimation using a particular state-space representation can even be more reasonable, e.g., when linearization is avoided.

Two research directions can be distinguished. The first problem is often referred to as Heterogeneous Track-to-Track Fusion (HT2TF) and describes problems where several tracks need to be fused. However, these tracks have different state-space representations, e.g., euclidean or polar coordinates, and there are often nonlinear transformations between state spaces. These transformations make the track-keeping of correlated estimation errors very complicated. The second class of problems describes the fusion of state estimates with overlapping or unequal state-space representation, which is often discussed for problems with large state spaces, e.g., the estimation of fine dust distributions [84].

The following section briefly reviews notable state-of-the-art methods for the fusion of state estimates from different local coordinate systems. Then, based on the current shortcomings of the related work, the previously introduced methods for distributed track-keeping are extended to state estimation problems with different local coordinate systems.

A Fusion of Heterogeneous State Estimates

In HT2TF problems, tracks from two or several moving targets that are subject to different state-space representations need to be fused. Unfortunately, these systems usually use different dynamic models, making the calculation of common information infeasible. The problem of HT2TF was first introduced by [25] for the track association problem, where several tracks with different state-space representations need to be checked whether they refer to the same moving target. The authors examine several track association methods with different assumptions and propose a robust track-to-track fusion method. In [137, 138], the authors propose a LMMSE and a ML approach to handle the fusion of heterogeneous state estimates. The papers show that both approaches can be successfully used to improve tracker performance and lead to comparable results as a central Kalman filter. However, since the authors did not calculate the real cross-covariance analytically, the fusion step is carried out by assuming the tracks are uncorrelated. The authors also approximate the cross-correlation in a steady-state case, but the approach shows no significant improvement.

Several papers propose the use of IMF, e.g., [90, 91, 135]. In [90], the authors explicitly model the common process noise identically to account for the fact that trackers observe the same target. The authors in [136] introduce an approach to calculate the cross-covariance matrix between heterogeneous estimates analytically. To cope with the nonlinear transformation between state-space representations, they propose to use the Jacobian to linearize the transformation. Their results show that using the cross-covariance matrix improves the fused estimates. The paper from [46] can be seen as a special case of HT2TF, where the local sensor nodes work in the full state space, but for the fusion, only a subset of the state space that is globally most valuable to the fusion is communicated. Therefore, the fusion of state estimates with state representation subject to linear or nonlinear transformations is common in many practical applications. However, the track-keeping of correlated estimation errors is complicated and usually infeasible in the case of nonlinear transformations. Nevertheless, there are also many applications with linear transformations, where track-keeping is possible and valuable for improving the fusion. Therefore, the extension of distributed track-keeping methods to this kind of problem is worthwhile.

B Fusion of Partially Overlapping State Estimates

There is an increasing number of applications using many spatially distributed sensor nodes, e.g., to monitor large-scale phenomena like traffic, power systems, or economics [63]. In order to reduce computational costs, there is a need to distribute computation to several instances. The authors in [73, 16] discuss approaches to decompose the system model to allow for distributed Kalman filtering. As discussed before, many methods have been designed to fuse estimates of the same state space, but they can often not be applied to problems with partially overlapping state estimates. A similar problem is called partition-based state estimation [43], where linear constrained systems are decomposed into smaller subsystems, but state estimates are not overlapping. In [43], the authors propose three moving horizon estimation algorithms that solve an optimization problem that scales with the number of states. In [123], a consensus method is proposed to fuse overlapping state estimates, and an empirical method is proposed in [119]. The authors in [99] formulate the fusion of partially overlapping state estimates as a WLS problem. However, because information about the cross-covariance is not available, the authors use CI to fuse estimates with unknown correlation. Furthermore, an approach using CI and a smoothing algorithm was proposed by [84, 85].

However, the current state of the art does not contain a systematic approach to calculate cross-covariances in such systems and utilize them for fusion. Since the cross-covariance is usually beneficial for the fusion step, the following section extends the distributed track-keeping of correlated estimation errors to systems with partially overlapping state estimates.

3.3.1 Linearly Transformed State Spaces

Distributed track-keeping of correlated estimation errors enables the optimal fusion of state estimates. This also applies to estimates subject to arbitrary transformations into states of equal or lower dimensions. However, nonlinear transformations are problematic, as the calculation of common information is challenging, which is further discussed in Section 3.3.2. Therefore, the focus lies on linear transformations between state spaces that allow the WLS fusion of estimates, as previously discussed. The following section discusses a general linear transformation between local state spaces and the WLS fusion of local state estimates. Furthermore, a systematic approach to reconstruct cross-covariances in systems with linearly transformed state spaces is derived. This is the basis for extending the introduced square root decomposition and sample-based reconstruction of cross-covariances. Finally, this section ends in a short evaluation example for a linear HT2TF problem featuring three linear trackers in a subspace of the original state space. Some of the content discussed in this section is also published in [151].

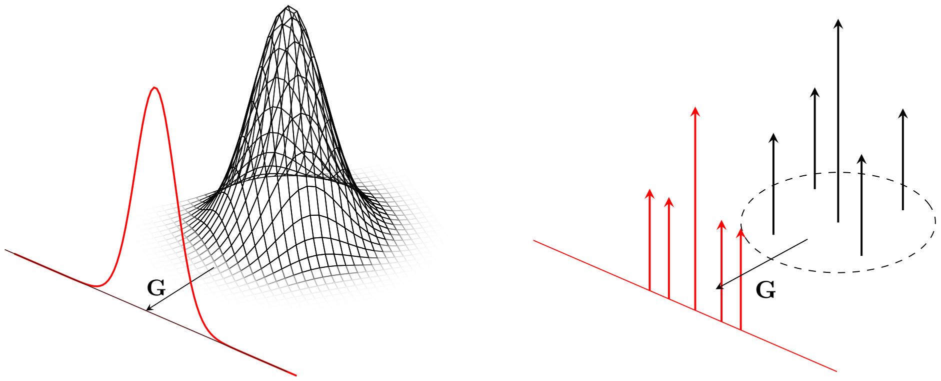
A States Estimates from Linearly Transformed State Spaces

The considered system is described by a linear time-invariant discrete system model similar to (2.1), where the state space is observed by L sensor nodes

$$\underline{z}_k^i = \mathbf{C}_g^i \cdot (\underline{\mathbf{x}}_k + \underline{t}_g^i) + \underline{\mathbf{v}}_k^i \,, \quad \text{with} \quad \underline{\mathbf{v}}_k^i \sim \mathcal{N}(\underline{0}, \mathbf{R}_k^i) \,.$$

Every sensor node i has a measurement model in the global state space with measurement matrix $\mathbf{C}_g^i$. It is assumed that measurements are subject to Gaussian distributed measurement noise $\underline{\mathbf{v}}_k^i$ with covariance matrix $\mathbf{R}_k^i$. Furthermore, $\underline{t}_g^i$ describes a sensor-specific offset from the origin of the global coordinate system to the origin O^i of the local coordinate system.

Instead of sending all measurements to a central processing unit, these measurements are processed by a local Kalman filter in every sensor node. An example for a system with several distributed trackers working in local state spaces can be seen in Figure 3.3. The local sensor nodes are



(a) Transformation of two-dimensional Gaussian into one-dimensional Gaussian.

(b) Transformation of samples in two-dimensional space into one-dimensional space.

Figure 3.2: Transformation from a two-dimensional to a one-dimensional state space (adapted from [151]).

estimating the state in a state space of the same or possibly smaller dimension $n_x^i \leq n_x$. The transformation from the global state space to the local state space is given by

$$\underline{\mathbf{x}}_k^i = \mathbf{G}^i \cdot (\underline{\mathbf{x}}_k + \underline{t}_g^i) = \mathbf{G}^i \underline{\mathbf{x}}_k + \underline{t}^i \,, \tag{3.11}$$

where the transformation $\mathbf{G}^i$ is a linear transformation from $\mathbb{R}^{n_x}$ into the linear Euclidean subspace $\mathbb{R}^{n_x^i}$. Since the local estimators use a different state-space representation, they employ a linearly transformed model of the global measurement model according to

$$\underline{z}_k^i = \mathbf{C}^i \underline{\mathbf{x}}_k^i + \underline{\mathbf{v}}_k^i \quad \text{with} \quad \underline{\mathbf{v}}_k^i \sim \mathcal{N}(\underline{0}, \mathbf{R}_k^i) \,, \tag{3.12}$$

with local measurement model $\mathbf{C}^i$. When the local state spaces are smaller than the global state space, the linear transformation maps the Gaussian distribution to a lower-dimensional Gaussian distribution. An example of such a linear transformation into a lower-dimensional state space can be seen in Figure 3.2, where the global state space is two-dimensional, but the local state space is one-dimensional. Therefore the two-dimensional Gaussian is transformed into a one-dimensional Gaussian distribution. In the case of nonlinear transformations between state spaces, the probability distribution might not be Gaussian anymore. This might require a different design of local estimators or the fusion step.

B Fusion of States Estimates from Linearly Transformed State Spaces

The following section introduces the fusion of estimates from linearly transformed state spaces. The most significant difference to the fusion of estimates from the same state space is that the linear transformation must be considered during the fusion. First, the unbiased estimate is constructed by

$$\hat{\underline{m}}_k^i = \hat{\underline{x}}_k^i - \underline{t}^i = \mathbf{G}^i \hat{\underline{x}}_k \,,$$

where the subtraction of $\underline{t}^i$ removes the offset between the different coordinate systems that result from (3.11). Afterward, the fusion can be formulated in the WLS sense

$$\underline{\hat{x}}_k^{\mathrm{WLS}} = \arg\min_{\underline{x}} \Big[\underline{\hat{m}}_k - \mathbf{G}\underline{x}_k\Big]^{\mathrm{T}} \mathbf{J}_k^{-1} \Big[\underline{\hat{m}}_k - \mathbf{G}\underline{x}_k\Big],$$

with joint state estimate $\underline{\hat{m}}_k = \Big[(\underline{\hat{m}}_k^i)^{\mathrm{T}} \dots (\underline{\hat{m}}_k^L)^{\mathrm{T}}\Big]^{\mathrm{T}}$. The matrix $\mathbf{G}$ accounts for the known transformations of global state space into the local state spaces

$$\mathbf{G} = \Big[(\mathbf{G}^1)^{\mathrm{T}} \dots [(\mathbf{G}^L)^{\mathrm{T}}\Big]^{\mathrm{T}} .$$

Furthermore, the joint covariance matrix is defined as

$$\mathbf{J}_k = \mathrm{E}\Big[(\underline{\hat{m}}_k - \mathbf{G}\underline{x}_k)(\underline{\hat{m}}_k - \mathbf{G}\underline{x}_k)^{\mathrm{T}}\Big] .$$

This joint covariance matrix accounts for the correlation of the local state estimates. Since state estimates from the local sensor nodes include common process noise and common prior information, the cross-covariances on the off-diagonals are nonzero.

The solution to this WLS problem can be formulated similarly to BSC-formulas in (2.10) by calculating a gain matrix

$$\mathbf{F} = \Big(\mathbf{G}^{\mathrm{T}}\mathbf{J}_k^{-1}\mathbf{G}\Big)^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{J}_k^{-1} .$$

Finally, the fusion rule for estimates from linearly transformed local state estimates is

$$\begin{aligned}\underline{\hat{x}}_k^{\mathrm{f}} &= \mathbf{F}\underline{\hat{m}}_k = \mathbf{P}_k^{\mathrm{f}}\mathbf{G}^{\mathrm{T}}\mathbf{J}_k^{-1}\underline{\hat{m}}_k ,\\ \mathbf{P}_k^{\mathrm{f}} &= \Big(\mathbf{G}^{\mathrm{T}}\mathbf{J}_k^{-1}\mathbf{G}\Big)^{-1} .\end{aligned}$$

Therefore, knowing the local linear transformations, the fusion rule is very similar to the standard WLS fusion (see (2.12) and (2.11)). However, to use it, cross-covariances are needed for the construction of the joint covariance matrix. The following section will be concerned with the reconstruction of the cross-covariances.

C Distributed Track-Keeping for Linearly Transformed State Spaces

In the following section the calculation of the cross-covariance matrix for linearly transformed state-space representations is derived. It is assumed, that the local state estimate $\underline{\hat{x}}^i$ is propagated during the prediction step using a local system model $\mathbf{A}^i = \mathbf{G}^i\mathbf{A}(\mathbf{G}^i)^{\mathrm{T}}$. Then, the cross-covariance after the prediction step between state estimates $\underline{\hat{x}}^i$ and $\underline{\hat{x}}^j$ is calculated by

$$\begin{aligned}\mathbf{P}_{k|k-1}^{i,j} &= \mathrm{E}[(\mathbf{A}_k^i\underline{\hat{x}}_{k-1|k-1}^i - \mathbf{G}^i\underline{\mathbf{x}}_k)(\mathbf{A}_k^j\underline{\hat{x}}^j - \mathbf{G}^j\underline{\mathbf{x}}_k)^{\mathrm{T}}]\\ &= \mathrm{E}\Big[(\mathbf{A}_k^i\underline{\hat{x}}_{k-1|k-1}^i - \mathbf{G}^i(\mathbf{A}_k\underline{\mathbf{x}}_{k-1} + \underline{\mathbf{w}}_k))(\mathbf{A}_k^j\underline{\hat{x}}_{k-1|k-1}^j - \mathbf{G}^j(\mathbf{A}_k\underline{\mathbf{x}}_{k-1} + \underline{\mathbf{w}}_k))\Big]\\ &= \mathbf{A}_k^i\mathbf{P}_{k-1|k-1}^{i,j}(\mathbf{A}_k^j)^{\mathrm{T}} + \mathbf{G}^i\mathbf{Q}_k(\mathbf{G}^j)^{\mathrm{T}} .\end{aligned}$$

Common process noise is determined by the parts of the state space that are affected by the same process noise and same prior information. Therefore, the correlation of estimation errors now

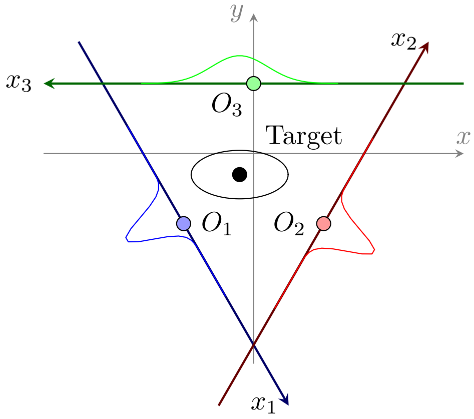


Figure 3.3: Problem sketch for a heterogeneous track-to-track fusion with three local filters estimating linear subsystems of the global state space (from [151]).

depends on the local transformation $\mathbf{G}^i$ and $\mathbf{G}^j$ that determine how much of the process noise $\mathbf{Q}_k$ is shared. As before, the cross-covariance is updated during the local measurement updates

$$\mathbf{P}^{i,j}_{k|k} = (\mathbf{I} - \mathbf{K}^i_k \mathbf{C}^i_k)\mathbf{P}^{i,j}_{k|k-1}(\mathbf{I} - \mathbf{K}^j_k \mathbf{C}^j_k)^{\mathrm{T}} \,.$$

With exception to the newly incorporated local transformation matrices, this calculation of the cross-covariance matrix is identical to the previously discussed approach. Therefore, the recursive calculation can also be rewritten as a sum of square root decompositioned noise covariances

$$\begin{aligned}\mathbf{P}^{i,j}_{k|k} &= \mathbf{T}^i_{0,k}\mathbf{G}^i\sqrt{\mathbf{P}_0}(\sqrt{\mathbf{P}_0})^{\mathrm{T}}(\mathbf{G}^j)^{\mathrm{T}}(\mathbf{T}^j_{0,k})^{\mathrm{T}} + \sum_{\tau=1}^{k}\mathbf{T}^A_{\tau,k}\mathbf{G}^i\sqrt{\mathbf{Q}_\tau}(\sqrt{\mathbf{Q}_\tau})^{\mathrm{T}}(\mathbf{G}^j)^{\mathrm{T}}(\mathbf{T}^j_{\tau,k})^{\mathrm{T}} \\ &= \sum_{\tau=0}^{k}\mathbf{\Sigma}^i_{\tau,\mathbf{Q}}(\mathbf{\Sigma}^j_{\tau,\mathbf{Q}})^{\mathrm{T}} \,,\end{aligned}$$

similar to Section 3.2.1. The only difference is that the mapping matrix $\mathbf{G}$ needs to be included in the creation of the local square root matrices. Following this discussion, the extension of the square root decomposition is as follows. At time step $k = 0$, the square root matrix is initialized with

$$\mathbf{S}^i_{0,\mathbf{Q}} = \mathbf{\Sigma}^i_{k,\mathbf{Q}} = \mathbf{G}^i\sqrt{\mathbf{P}_0} \,.$$

This matrix is linearly transformed during the prediction step, and a new noise term $\mathbf{\Sigma}^i_{k,\mathbf{Q}} = \mathbf{G}^i\sqrt{\mathbf{Q}}$ is included. The matrix is then updated using the gain matrix of the Kalman filter update $\mathbf{L}^i_k = \mathbf{I} - \mathbf{K}^i_k\mathbf{C}^i_k$

$$\mathbf{S}^i_{k,\mathbf{Q}} = \mathbf{L}^i_k\left[\mathbf{A}^i_k\mathbf{S}^i_{k-1,\mathbf{Q}} \,,\ \mathbf{\Sigma}^i_{k,\mathbf{Q}}\right] .$$

Thus, the remaining fusion algorithm is executed as described in Section 3.2.1.

D Extension to the Sample-Based Reconstruction

As discussed before, the common information depends on the common parts of the local state space of node i and j. Therefore, state estimates contain dependent information where they are affected by the same process noise or common prior information, thus, contributing to the cross-covariance matrix. This behavior of correlated and uncorrelated parts can be modeled by creating samples that contain independent and dependent sections. This concept of correlated and uncorrelated samples for fusion is also discussed in [115]. Here, the correlated and uncorrelated sampling is achieved by the sampling scheme with the characteristics described in (3.6).

As before, a block diagonal matrix (see (3.7)) is created

$$\mathbf{D}^i = \mathrm{blkdiag}\left(\mathbf{G}^i\sqrt{\mathbf{P}_0}, \mathbf{G}^i\sqrt{\mathbf{Q}_1}, \ldots, \mathbf{G}^i\sqrt{\mathbf{Q}_{\mathcal{T}}}\right) .$$

However, this time it is extend with the mapping matrix $\mathbf{G}^i$ to account for the transformation into the subspace of node i. Then, the local subspace sample set similar to (3.8) is obtained

$$\begin{aligned} \underline{d}_k^{i,m} &= \mathbf{D}^i \underline{p}^m \quad , \quad \forall m = 1, \ldots, M \\ &= \left[(\underline{s}_{k|k}^{i,m})^{\mathrm{T}}, \quad (\underline{w}_{k+1}^{i,m})^{\mathrm{T}} \quad , \ldots , \quad (\underline{w}_{k+\mathcal{T}}^{i,m})^{\mathrm{T}} \right]^{\mathrm{T}} . \end{aligned}$$

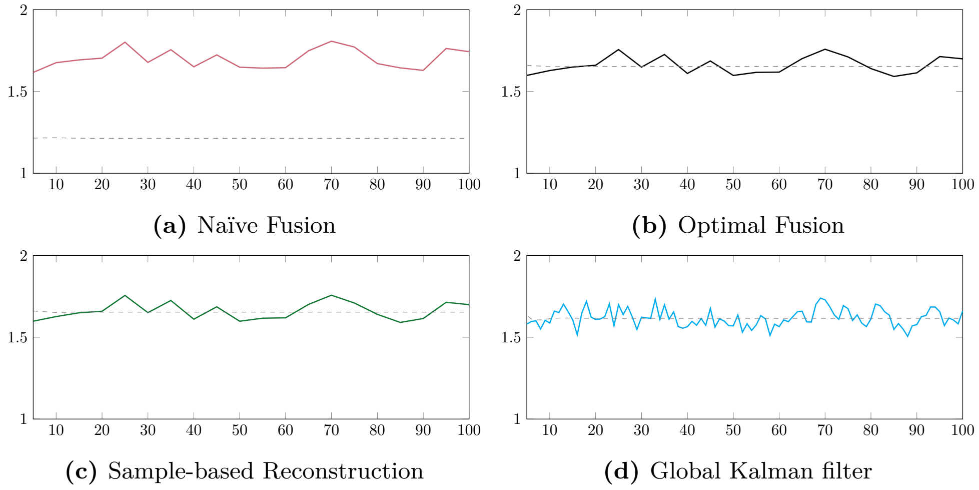
Because of the local transformation, the process noise sample sets are not identical in every sensor node. Therefore, every sensor node now has its own set of process noise samples that only correlate with sample sets where the state estimates are affected by the same process noise and common prior information depending on the local transformations. The number of samples M is equal to the number of samples when sampling in the global state space, but the dimension of the local sample sets are reduced to the dimension of the local state spaces n_x^i and n_x^j.

The idea of this construction is that only the local transformation matrices $\mathbf{G}^i$ and $\mathbf{G}^j$ determine the common parts of the state spaces of the local estimators. The local sample sets are processed through the local Kalman filter by the prediction step (3.9) and the measurement update (3.9). When the fusion step occurs, the cross-covariance matrix can be calculated according to (3.10).

E Evaluation Using Tracking Example with Three Local Filter Estimating in Linear Subsystems of the Global State Space

The following evaluation example is taken from [151]. This evaluation example shows that optimal track-keeping of correlated estimation errors is possible in systems where every local tracker only possesses incomplete local knowledge from a subspace. Furthermore, the proposed fusion approach can fully construct a credible global estimate. We consider a target moving on a two-dimensional plane where the motion can be described by a constant velocity model with time constant $\Delta T = 0.1$. Moreover, the position of the target in two dimensions is denoted by $\xi = [\xi_x, \xi_y]^{\mathrm{T}}$ and the translational velocity in two dimensions by $\nu = [\nu_x, \nu_y]^{\mathrm{T}}$

$$\begin{bmatrix} \xi_x \\ \xi_y \\ \nu_x \\ \nu_y \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \xi_x \\ \xi_y \\ \nu_x \\ \nu_y \end{bmatrix}_k + \underline{\mathbf{w}}_k .$$



(a) Naïve Fusion

(b) Optimal Fusion

(c) Sample-based Reconstruction

(d) Global Kalman filter

Figure 3.4: Comparison of the trace of the fused covariance matrix (dashed) and actual mean squared error for 1000 MCR (adapted from [151]).

The system is affected by additive white Gaussian noise $\underline{\mathbf{w}}_k$ with noise power $q_1 = 2$, $q_2 = 0.5$ according to

$$\mathbf{Q} = \begin{bmatrix} q_1\frac{1}{3}\Delta T^3 & 0 & q_1\frac{1}{2}\Delta T^2 & 0 \\ 0 & q_2\frac{1}{3}\Delta T^3 & 0 & q_2\frac{1}{2}\Delta T^2 \\ q_1\frac{1}{2}\Delta T^2 & 0 & q_1\Delta T & 0 \\ 0 & q_2\frac{1}{2}\Delta T^2 & 0 & q_2\Delta T \end{bmatrix} .$$

The target is tracked by three local estimators that are arranged in an equilateral triangle (see Figure 3.3). The local estimators only estimate a two dimensional subspace of the global state space. Therefore, they only estimate the position and the velocity along a single axis that is the x-axis of the original system rotated by an angle ϕ_i

$$\phi_1 = -\pi/3\,,\ \phi_2 = \pi/3\,,\ \phi_3 = -\pi\,.$$

The origins O_i of the new coordinate systems have an offset $\underline{t}_g^i$ to the original euclidean coordinate system according to

$$\underline{t}_g^1 = \begin{bmatrix}-5 & 5 & 0 & 0\end{bmatrix}^{\mathrm{T}},\ \underline{t}_g^2 = \begin{bmatrix}5 & -5 & 0 & 0\end{bmatrix}^{\mathrm{T}},\ \underline{t}_g^3 = \begin{bmatrix}0 & 5 & 0 & 0\end{bmatrix}^{\mathrm{T}}.$$

The linear transformation from the full state space to the sub state spaces can be described by equation (3.11), where the linear transformation is given by

$$\mathbf{G}^i = \begin{bmatrix} \cos(\phi_i) & -\sin(\phi_i) & 0 & 0 \\ 0 & 0 & \cos(\phi_i) & -\sin(\phi_i) \end{bmatrix} .$$

Finally, the local measurement model is given by (3.12), where the measurements are drawn from the global state space by equation (2.2)

$$\mathbf{C}_g^i = \begin{bmatrix}\cos(\phi_i) & -\sin(\phi_i) & 0 & 0\end{bmatrix},\ \mathbf{C}^i = \begin{bmatrix}1 & 0\end{bmatrix},$$

with the local measurement noise matrices

$$\mathbf{R}^1 = 1\,,\ \mathbf{R}^2 = 0.5\,,\ \mathbf{R}^3 = 0.25\,.$$

Every local tracker uses a local system model $\mathbf{A}^i = \mathbf{G}^i\mathbf{A}(\mathbf{G}^i)^{\mathrm{T}}$ that is a result of the linear transformation $\mathbf{G}^i$ (see Section 3.3.1.C). This local models describes the motion of the target as a constant velocity model, but in a one-dimensional space

$$\begin{bmatrix}\xi_x\\ \nu_x\end{bmatrix}_{k+1}^{i} = \begin{bmatrix}1 & \Delta T\\ 0 & 1\end{bmatrix}\begin{bmatrix}\xi_x\\ \nu_x\end{bmatrix}_{k}^{i} + \underline{\mathbf{w}}_k^i\,.$$

The proof that the model is still a constant velocity model but in lower-dimensional state space and that process noise matrices of the local trackers are different from each other and depend on the transformation was shown in [151].

In this example, the sample-based reconstruction method is used. Moreover, the sampling is executed using UKF samples [68], but the sample in the center of the distribution has double the weight of the other samples. The fusion step is carried out every 5th time step, which results in a time horizon $\mathcal{T} = 6$ for the sample set, which also includes the initial covariance matrix. The fusion result is compared with the naïve fusion, which neglects the cross-covariance, and with the optimal reconstruction that uses the recursive formula of (3.1) and (3.2) to reconstruct the cross-covariance optimally. Finally, the fusion result is compared with the result of the central Kalman filter that is given all measurements and estimates the state in the full state space.

Figure 3.4 shows the mean squared error of the fused state estimates and the trace of the fused covariance matrix. The global Kalman filter yields the best results, which is expected since it uses the measurements more efficiently then the optimal fusion. The optimal fusion and the sample-based reconstruction yield the same results. Therefore, the distributed track-keeping is identical to the centralized track-keeping as intended. The results also show that neglecting the correlations leads to a higher error, as the naïve fusion is doing significantly worse than the other methods. Furthermore, the estimated error of the fused covariance matrix of the global Kalman filter (Figure 3.4d), the optimal fusion (Figure 3.4b) and the proposed approach (Figure 3.4c) matches the actual error. Naïve fusion (Figure 3.4a) results in an overconfident covariance matrix that does not match the real error and therefore leads to a noncredible tracker.

3.3.2 Overlapping State Estimates

A very similar problem to the fusion of state estimates from linearly transformed states-spaces is the fusion of state estimates that are partially overlapping. This problem is also discussed in [147], where the sample-based reconstruction was adopted and a more complicated evaluation example was considered.

The idea is, that a large-scale phenomenon, e.g., a heat distribution, is estimated. Instead of sending a vast amount of measurement data to a central Kalman filter, the estimation problem is decomposed into smaller estimation problems that are solved in local Kalman filters. These local estimators observe only a small subset of states, which results in state estimates that are partially overlapping.

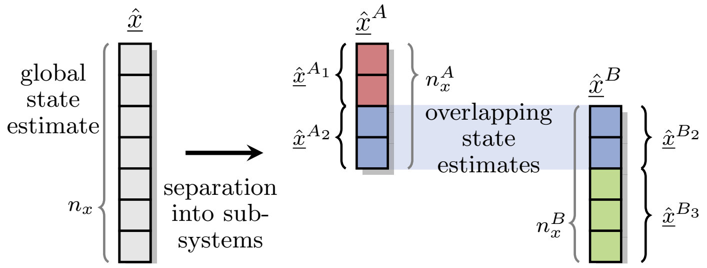

$\hat{\underline{x}}$
global state estimate
n_x
separation into sub-systems
$\hat{\underline{x}}^A$
$\hat{\underline{x}}^{A_1}$
$\hat{\underline{x}}^{A_2}$
n_x^A
overlapping state estimates
$\hat{\underline{x}}^B$
$\hat{\underline{x}}^{B_2}$
$\hat{\underline{x}}^{B_3}$
n_x^B

Figure 3.5: Separation of the global state estimate into the two overlapping subsystems A and B with state estimates (from [147]).

This section follow the WLS approach proposed in [99]. Two local sensor nodes A and B are given with state estimates

$$\hat{\underline{x}}^A = \left[(\hat{\underline{x}}^{A_1})^{\mathrm{T}}, \quad (\hat{\underline{x}}^{A_2})^{\mathrm{T}}\right]^{\mathrm{T}} \quad \text{and} \quad \hat{\underline{x}}^B = \left[(\hat{\underline{x}}^{B_2})^{\mathrm{T}} \quad (\hat{\underline{x}}^{B_3})^{\mathrm{T}}\right]^{\mathrm{T}}$$

that overlap at sections A_2 and B_2. A sketch of this problem is shown in Figure 3.5. Thus, the global system (see equations (2.1) and (2.2)) is separated into subsystems so that the local system model is rewritten as

$$\underline{\mathbf{x}}_k^i = \mathbf{A}_k^i \underline{\mathbf{x}}_{k-1}^i + \underline{\mathbf{w}}_k^i \quad \text{with} \quad \underline{\mathbf{w}}_k^i \sim \mathcal{N}(\underline{0}, \mathbf{Q}_k^i)\,,$$

where the dimension of the local subsystem n_x^i is smaller than the dimension n_x of the global system.

Each state estimate can be seen as an observation [99] with measurement matrix $\mathbf{H}$ (see equation (2.11)), which determines how the local state estimates map into the global state space

$$\begin{bmatrix} \underline{\mathbf{x}}_{A_1} \\ \underline{\mathbf{x}}_{A_2} \\ \underline{\mathbf{x}}_{B_2} \\ \underline{\mathbf{x}}_{B_3} \end{bmatrix} = \mathbf{H}\underline{\mathbf{x}} + \tilde{\underline{\mathbf{x}}}\,, \quad \text{with} \quad \mathbf{H} := \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}0\\0\end{matrix}}\right\} \mathbf{H}^A \\ \left.\vphantom{\begin{matrix}0\\0\end{matrix}}\right\} \mathbf{H}^B \end{matrix}\,,$$

where the matrices $\mathbf{H}^A$ and $\mathbf{H}^B$ determine, which part of the state space are occupied by the local state estimates $\hat{\underline{x}}^A$ and $\hat{\underline{x}}^B$. Finally, the term $\tilde{\underline{\mathbf{x}}}$ denotes the measurement error and has covariance

$$\mathbf{J} = \begin{bmatrix} \mathbf{P}^A & \mathbf{P}^{A,B} \\ \mathbf{P}^{B,A} & \mathbf{P}^B \end{bmatrix} = \left[\begin{array}{cc:cc} \mathbf{P}^{A_1} & \mathbf{P}^{A_1,A_2} & \mathbf{P}^{A_1,B_2} & \mathbf{P}^{A_1,B_3} \\ \mathbf{P}^{A_2,A_1} & \mathbf{P}^{A_2} & \mathbf{P}^{A_2,B_2} & \mathbf{P}^{A_2,B_3} \\ \hdashline \mathbf{P}^{B_2,A_1} & \mathbf{P}^{B_2,A_2} & \mathbf{P}^{B_2} & \mathbf{P}^{B_2,B_3} \\ \mathbf{P}^{B_3,A_1} & \mathbf{P}^{B_3,A_2} & \mathbf{P}^{B_3,B_2} & \mathbf{P}^{B_3} \end{array}\right]$$

that is equivalent to the joint covariance matrix from equation (2.8). Therefore, the distributed track keeping for the system with overlapping state estimates is similar to the linearly transformed systems in the previous section. Here, the linear transformation is determined by the mapping matrix $\mathbf{H}$ that describes how the local estimates merge into the global estimate.

A Discussion

In theory, the separation of high-dimensional state space into smaller subsystems can be beneficial as it helps to save computational power and communication bandwidth. However, it requires some care during the design of the local estimators. The reason is that the process noise affects the whole system, but applying an arbitrary transformation that maps the global system into lower-dimensional state space cannot accurately model process noise in the local estimators. Therefore, the uncertainty of the local estimates is underestimated, leading to estimates that are not credible.

This problem is also discussed by the authors in [90]. Here, the authors state that the dimension of the state spaces of the local estimates needs to be equal. Furthermore, there must be a one-to-one mapping between the state spaces. Moreover, the same process noise needs to be incorporated into the local estimates since all local trackers estimate the same target.

Unfortunately, this is not the case for many evaluation examples referred to when considering overlapping state estimates or HT2TF problems. HT2TF applications usually consider an active tracker that operates in a state space of full dimension and is then fused with an estimate of a passive tracker that often works in a lower-dimensional space, e.g., only tracks the angle or angle velocity [131, 136]. Therefore, the process noise is not identical in the active and the passive tracker because of the information loss due to transformation. Hence, the state spaces cannot be mapped one-to-one. In the overlapping estimation problem in [119, 99], the authors consider the estimation of a heat distribution in a rod. Here, the process noise is modeled to affect the complete rod. However, the same problems appear by cutting the rod into smaller sections, leading to underestimating the uncertainty.

Another critical aspect is that the system models considered in the local estimators are often nonlinear. Even if the trackers are affected by the same process noise, the process noise assumed during the prediction step differs because of the nonlinear transformation that depends on the current state. If the nonlinearities are not severe, the process noise could be assumed identical, e.g., if the target is far away. With this knowledge, the square root decomposition could be used as well. The sample-based reconstruction has the most impact if nonlinearities are severe. However, this leads to the problem of state-dependent process noise. In applications where this is not a problem, the sampling technique has to be adapted to catch the probability distribution after the transformation, meaning that the sample size or the sampling scheme needs to be adapted.

B Evaluation Example for the Fusion of Overlapping State Estimates

The problem of transforming the global model into a lower-dimensional state-space model is highlighted in the following section. First, it is shown that an insufficient separation of the system model leads to inconsistent fused estimates. Furthermore, it is shown that the local estimation has to be done in full state space in these cases. However, it is possible to fuse a smaller subset of the local state space to obtain credible fusion results, and the correlation of estimation errors can be tracked in a distributed fashion using the proposed approach.

The state of a target moving on a straight line is estimated. The 1-dimensional constant velocity model is

$$\mathbf{A} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \quad , \quad \begin{bmatrix} \frac{\Delta T^3}{3} & \frac{\Delta T^2}{2} \\ \frac{\Delta T^2}{2} & \Delta T \end{bmatrix} \quad , \quad \Delta T = 0.1\,.$$

Two sensor nodes A and B are given, and every sensor node has a linear measurement model with measurement covariance matrix according to

$$\mathbf{C}^A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad , \quad \mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & 0.5^2 \end{bmatrix} \quad , \quad \mathbf{C}^B = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad , \quad \mathbf{R} = 0.05^2\,.$$

Sensor node A measures both states and sensor node B only measures the first state estimate. Therefore it is decided that node B only estimates the state it observes, and the estimate $\underline{\hat{x}}^A$ is only fused with the first state of the estimate $\underline{\hat{x}}^B$. A naïve separation of the state space is conducted, where the new system and measurement model of node B is transformed to

$$\mathbf{A}^B = 1 \quad , \quad \mathbf{Q}^B = \frac{\Delta T^3}{3}\,.$$

Because only the first state of $\underline{\hat{x}}^B$ is fused into $\underline{\hat{x}}^A$ the mapping matrices are given by

$$\mathbf{H}^A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad , \quad \mathbf{H}^B = \begin{bmatrix} 1 & 0 \end{bmatrix}\,.$$

The fusion is carried out at every 10th time step, and the fused estimate is used to reinitialize the state estimate. In the case where the node only estimates the subspace of the state space, this reinitialization step is

$$\mathbf{P}^B = \mathbf{H}^B \mathbf{P}^{\mathrm{f}} (\mathbf{H}^B)^{\mathrm{T}}\,.$$

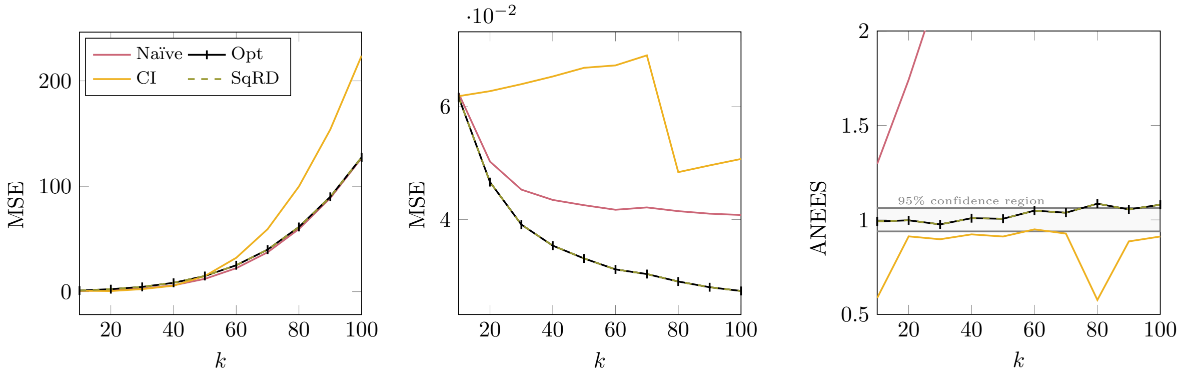
Correlated estimation errors are kept track of using the square root decomposition (SqRD) approach. Figure 3.6a shows the MSE of the proposed fusion approach. All fusion methods result in inconsistent estimates that diverge quickly. Even when node A observes two states and node B does not even observe the state left out of the estimation, the separation of this system does not yield functioning estimators.

In the light of the previous results, it is now assumed that the estimators work in full state space. However, the fusion is still carried out with $\underline{\hat{x}}^A$ in full state space and $\underline{\hat{x}}^B$ fusing only its first state. The track-keeping is still executed using SqRD, but the correlation is only needed for the first state of $\underline{\hat{x}}^B$. Thus, the square root matrix is transformed to the subspace before the communication by

$$\mathbf{S}^B_{k,\mathbf{Q}} := \mathbf{H}^B \mathbf{S}^B_{k,\mathbf{Q}}\,.$$

Then the cross-covariance matrix is reconstructed as already discussed in Section 3.2.1.

Figure 3.6b shows that the fusion result does not diverge when the local state estimation is performed in full state space and then transformed afterward. It can also be seen that the SqRD is identical to the result of the optimal centralized reconstruction and that the optimal fusion yields the best fusion results. Figure 3.6c further shows that the optimal fusion result is consistent



(a) MSE using estimators in subspace.

(b) MSE using estimators in full state space.

(c) ANEES using estimators in full state space.

Figure 3.6: Comparison of fusion results for overlapping state estimates with estimation in subspace and estimation in full state space.

and stays within the 90% confidence interval, while the fusion result of CI is conservative and the naïve fusion result underestimates the error significantly.

The results of this sections show that the distribution in local state spaces has to be performed carefully as process noise may affect the complete state space. Therefore, a naïve separation of states can be problematic and will result in local estimators that are not credible. Moreover, the local estimation needs to be performed in a sufficiently big state space to capture the complete uncertainty of the system model.

3.3.3 Conclusion to Reconstruction of Cross-Covariances for Estimates from Different Coordinate Systems

This section discussed the similarities between the fusion of partially overlapping state estimates and the HT2TF problem. We derived an approach to linearly transform state estimates into local state spaces with the same or lower dimensions than the global state space. The derived method is used to keep track of correlated estimation errors using the square root decomposition or the sample-based reconstruction of correlated noise covariances. Both methods are identical to the centralized track-keeping of cross-covariances. The results also show that it requires proper care to design distributed estimation applications with different local state-space representations. Whenever the dimension of the system is reduced, the uncertainty of the global system might be underestimated, leading to inconsistent local trackers and, therefore, inconsistent fusion results.

3.4 Fully Decentralized Track-Keeping of Cross-Covariances

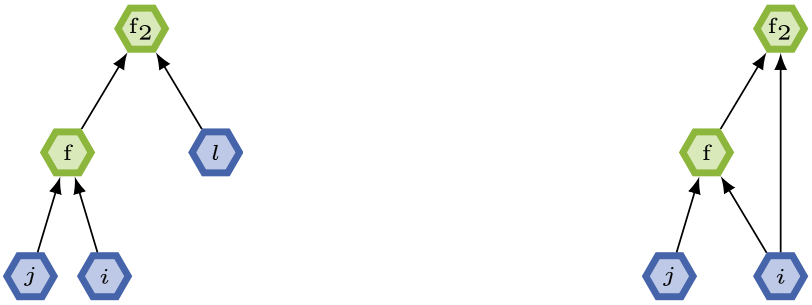
Sensor networks often contain a large number of sensor nodes that can be spatially distributed over large areas. However, energy storage and communication bandwidth are limited in many sensor nodes, making it impossible to send all information to one central fusion node over a significant distance. Instead, sensor nodes often send information over several hops until it finally reaches its destination point within the sensor network. In addition, there is no central fusion center in some sensor networks, but instead, sensor nodes are autonomous and require only the support of neighboring sensor nodes.

The following sections consider sensor networks without a single dedicated fusion center, which has been discussed in Section 2.1.2. Instead, the fusion of state estimates is either done in a hierarchical topology using intermediate fusion steps or in a fully decentralized network topology where every sensor node can execute the fusion step independently. In these networks, the track-keeping of correlated estimation errors depends on the information flow through the network, which is hard to keep track of as no central node is present.

Many methods have been proposed for sensor networks with hierarchical or fully decentralized network topologies. CI [67, 27, 111] is still widely used because it is guaranteed to stay consistent for any possible correlation between the estimation errors. However, CI is often overly pessimistic since the information that is passed through sensor nodes is usually not fully correlated. Moreover, the local estimates include information exclusive to one sensor node, e.g., because it includes measurements that are not shared with any other sensor node. Several approaches are based on the information filter approach, e.g., the information graph [30] to keep track of previous communication paths. The channel filter [57] can also use this knowledge to filter out previously communicated information and, similarly, IMF [28, 131] works on the same principle. When the fusion is executed at full rate, meaning that a fusion step is executed after every processing step, these methods are optimal. Nevertheless, these approaches work only approximately because of the incorporation of process noise or when the full rate is not fulfilled. However, these algorithms perform well in many applications, e.g., if the process noise covariance matrix is relatively small.

Another class of algorithms aims to converge to a global estimate. These methods iteratively exchange information between neighboring nodes until a consensus is reached. Example of such methods are consensus on measurements [104], consensus on information [15, 105], or hybrid approaches [17, 87]. Consensus methods can be regarded as suboptimal fusion rules [31] where averaging the information does not represent the actual information in the network and does not consider redundant information systematically. A similar method to consensus methods are the diffusion methods [62, 22]. Diffusion methods are better suited for highly dynamic problems because they do not wait until a consensus is reached [31].

These methods either try to approximate the correlated information using bounds, or they try to identify and fuse only new information. However, none of these methods uses the actual cross-covariance for the optimal fusion. The reason is that keeping track of correlated estimation errors is cumbersome in these networks. However, the reconstruction of cross-covariance is advantageous, as it allows optimal fusion with consistent fusion results that are generally more accurate and do not over- or underestimate the uncertainty. Therefore, this section proposes an extension to the distributed track-keeping using samples and square roots to allow fusion in fully decentralized networks. First, track-keeping for hierarchical fusion is investigated. Afterward, a solution for fully decentralized network topologies that are subject to information loops is proposed.



(a) Hierarchical fusion. **(b)** Decentralized fusion with information loop.

Figure 3.7: Two decentralized network topologies, sensor nodes in blue and fusion nodes in green.

3.4.1 Optimal Reconstruction for Hierarchical Network Topologies

In hierarchical fusion topologies, a subset of sensor nodes fuses their estimates and that fused estimate is passed on to the hierarchically next layer, where it is subsequently fused with other state estimates again. This process is repeated until the hierarchically highest fusion node is reached. Because of the intermediate fusion steps, the correlation of estimation errors changes constantly. Therefore, this information must be incorporated in the additional information used to reconstruct the cross-covariances to allow correct reconstruction.

The network topology depicted in Figure 3.7a is assumed, where three sensor nodes i, j and l are present. First, node i fuses its estimate with an estimate received from node j by using fusion formulas (2.6) and (2.7). The required cross-covariance matrices $\mathbf{P}^{i,j} = (\mathbf{P}^{j,i})^{\mathrm{T}}$ are obtained by the square root decomposition, i.e., by using (3.5). Afterward, the intermediate fusion result $\underline{\hat{x}}^{\mathrm{f}}$, that contains information from node i and j, is fused with the state estimate from node l.

The cross-covariance accounting for the correlation of estimation errors between $\underline{\hat{x}}^{\mathrm{f}}$ and $\underline{\hat{x}}^{l}$ is constructed by

$$\begin{aligned}\mathbf{P}^{\mathrm{f},l} &= \mathrm{E}[(\underline{\hat{x}}^{\mathrm{f}} - \underline{\mathbf{x}})(\underline{\hat{x}}^{l} - \underline{\mathbf{x}})^{\mathrm{T}}] \\ &= \mathrm{E}[(\mathbf{F}^{i}\,\underline{\hat{x}}^{i} + \mathbf{F}^{j}\,\underline{\hat{x}}^{j} - \underline{\mathbf{x}})(\underline{\hat{x}}^{l} - \underline{\mathbf{x}})^{\mathrm{T}}] \\ &= \mathbf{F}^{i}\,\mathbf{P}^{i,l} + \mathbf{F}^{j}\,\mathbf{P}^{j,l}\,.\end{aligned}$$

The dependencies $\mathbf{P}^{i,l}$ and $\mathbf{P}^{j,l}$ are given by the corresponding square root decompositions, i.e.,

$$\mathbf{P}^{i,l} = \mathbf{S}^{i}_{\mathbf{Q}}\big(\mathbf{S}^{l}_{\mathbf{Q}}\big)^{\mathrm{T}} \text{ and } \mathbf{P}^{j,l} = \mathbf{S}^{j}_{\mathbf{Q}}\big(\mathbf{S}^{l}_{\mathbf{Q}}\big)^{\mathrm{T}}\,,$$

where the fused square root decomposition for the reconstruction of $\mathbf{P}^{\mathrm{f},l}$ has the form

$$\mathbf{S}^{\mathrm{f}}_{\mathbf{Q}} = \mathbf{F}^{i}\mathbf{S}^{i}_{\mathbf{Q}} + \mathbf{F}^{j}\mathbf{S}^{j}_{\mathbf{Q}}\,, \tag{3.13}$$

Therefore, the cross-covariance matrix $\mathbf{P}^{\mathrm{f},l} = \mathbf{S}^{\mathrm{f}}_{\mathbf{Q}}(\mathbf{S}^{l}_{\mathbf{Q}})^{\mathrm{T}}$ can be reconstructed for any node l. The same approach can be used to achieve hierarchical fusion using the sample-based reconstruction. Here, a sample set $\{\underline{s}^{\mathrm{f}}\}_{m=1}^{M}$ is constructed

$$\underline{s}^{\mathrm{f},m} = \mathbf{F}^{i}\,\underline{s}^{i,m} + \mathbf{F}^{j}\,\underline{s}^{j,m} \quad , \quad \forall m = 1, \ldots, M\,.$$

The hierarchical fusion is vital for many network topologies, including fully decentralized ones, as intermediate fusion steps also occur there. Therefore, the knowledge from this section helps to extend the framework to other topologies as well.

3.4.2 Optimal Reconstruction for Fully Decentralized Network Topologies

Fusion in decentralized network topologies is very challenging because there is no hierarchy to ensure the track-keeping of information. Furthermore, information loops frequently occur that are hard to keep track of. However, many networks are decentralized by design, as they are more robust, scalable, and can be better adapted to different tasks. Moreover, decentralized sensor networks can have sensor nodes that are autonomous and therefore only take care of their task. Therefore, decentralized sensor networks require a robust and versatile method to keep track of correlated estimation errors in a distributed fashion that accounts for previously incorporated information.

In decentralized sensor networks, sensor nodes also suffer from correlated estimation errors due to common prior information and process noise, resulting from estimating the same target. However, because of information loops, previously incorporated measurements are also shared and can be reintroduced to a node, leading to correlated measurement information. The network topology depicted in Figure 3.7b is assumed, where two sensor nodes i, j are present, but the result of the fusion step is fused again with the estimate of node i. The cross-covariance between the fused estimate and the estimate of node i is

$$\begin{aligned}
\mathbf{P}^{\mathrm{f},i} &= \mathrm{E}\big[(\underline{\hat{x}}^{\mathrm{f}} - \underline{\mathbf{x}})(\underline{\hat{x}}^{i} - \underline{\mathbf{x}})^{\mathrm{T}}\big] \\
&= \mathrm{E}[(\mathbf{F}^{i}\,\underline{\hat{x}}^{i} + \mathbf{F}^{j}\,\underline{\hat{x}}^{j} - \underline{\mathbf{x}})(\underline{\hat{x}}^{i} - \underline{\mathbf{x}})^{\mathrm{T}}] \\
&= \mathbf{F}^{i}\,\mathbf{P}^{i,i} + \mathbf{F}^{j}\,\mathbf{P}^{j,i}\,.
\end{aligned}$$

The cross-covariance $\mathbf{P}^{j,i}$ can be calculated as discussed in section 3.4.1. The other cross-covariance matrix in this equation denoted $\mathbf{P}^{i,i}$ is the correlation of the estimate $\underline{\hat{x}}^{i}$ with itself. This covariance matrix can be calculated according to

$$\begin{aligned}
\mathbf{P}^{i}_{k|k} &= \mathrm{E}\Big[(\underline{\hat{x}}^{i}_{k|k} - \underline{\mathbf{x}}_k)(\underline{\hat{x}}^{i}_{k|k} - \underline{\mathbf{x}}_k)^{\mathrm{T}}\Big] \\
&= \mathrm{E}\Big[\big(\underline{\hat{x}}^{i}_{k|k-1} + \mathbf{K}^{i}_{k}\underline{y}^{i}_{k} - \underline{\mathbf{x}}_k\big)\big(\underline{\hat{x}}^{i}_{k|k-1} + \mathbf{K}^{i}_{k}\underline{y}^{i}_{k} - \underline{\mathbf{x}}_k\big)^{\mathrm{T}}\Big] \\
&= \mathrm{E}\Big[\big(\underline{\hat{x}}^{i}_{k|k-1} + \mathbf{K}^{i}_{k}(\underline{z}^{i}_{k} - \mathbf{C}^{i}_{k}\underline{\hat{x}}^{i}_{k|k-1})\big)\big(\underline{\hat{x}}^{i}_{k|k-1} + \mathbf{K}^{i}_{k}(\underline{z}^{i}_{k} - \mathbf{C}^{i}_{k}\underline{\hat{x}}^{i}_{k|k-1})\big)^{\mathrm{T}}\Big] \\
&= (\mathbf{I} - \mathbf{K}^{i}_{k}\mathbf{C}^{i}_{k})\mathrm{E}\Big[\big(\underline{\hat{x}}^{i}_{k|k-1} - \underline{\mathbf{x}}_k\big)\big(\underline{\hat{x}}^{i}_{k|k-1} - \underline{\mathbf{x}}_k\big)^{\mathrm{T}}\Big](\mathbf{I} - \mathbf{K}^{i}_{k}\mathbf{C}^{j}_{k})^{\mathrm{T}} + \mathbf{K}^{i}_{k}\mathrm{E}\Big[\underline{\mathbf{v}}^{i}_{k}(\underline{\mathbf{v}}^{i}_{k})^{\mathrm{T}}\Big](\mathbf{K}^{i}_{k})^{\mathrm{T}} \\
&= \mathbf{L}^{i}_{k}\mathbf{P}^{i}_{k|k-1}(\mathbf{L}^{i}_{k})^{\mathrm{T}} + \mathbf{K}^{i}_{k}\mathbf{R}^{i}_{k}(\mathbf{K}^{i}_{k})^{\mathrm{T}}\,.
\end{aligned}$$

The difference between the previously discussed calculation of the cross-covariance matrix to the calculation of the covariance matrix is the occurrence of the measurement covariance matrix $\mathbf{R}^{i}_{k}$ that is now also a possibly correlated covariance matrix. Therefore, additionally, the square root matrix has to keep track of correlated measurement information.

Consequently, a new square root matrix containing the measurement noise covariances is constructed

$$\mathbf{S}^{i}_{k,\mathbf{R}^i} = \Big[\mathbf{\Sigma}^{i}_{1,\mathbf{R}^i}, \mathbf{\Sigma}^{i}_{2,\mathbf{R}^i}, \ldots, \mathbf{\Sigma}^{i}_{k,\mathbf{R}^i}\Big]\,.$$

This square root matrix is initialized at time step $k = 1$ with

$$\mathbf{S}^i_{1,\mathbf{R}^i} = \boldsymbol{\Sigma}^i_{1,\mathbf{R}^i} = \mathbf{K}^i_1 \sqrt{\mathbf{R}^i_1}\,,$$

where $\mathbf{R}^i_1$ is the measurement covariance matrix of the first measurement (2.2) at time step 1 of node i, and where the matrix $\mathbf{K}^i_1$ is the Kalman gain used in this measurement update. This square root matrix is updated recursively whenever a new measurement is available at node i, and a new entry is appended to the matrix according to

$$\mathbf{S}^i_{k,\mathbf{R}^i} = \left[\mathbf{L}^i_k \mathbf{A}_k \mathbf{S}^i_{k-1,\mathbf{R}^i}\ ,\ \boldsymbol{\Sigma}^i_{k,\mathbf{R}^i}\right] \tag{3.14}$$

with

$$\boldsymbol{\Sigma}^i_{k,\mathbf{R}^i} = \mathbf{K}^i_k \sqrt{\mathbf{R}^i_k}\,.$$

In decentralized sensor networks, two nodes that want to fuse their estimates need to communicate all locally processed square root matrices: one for the common prior information and common process noise, and one for every sensor node, including itself, that they previously received information from. While the square root matrices for the common prior information and common process noise can be merged during the fusion step as explained in Section 3.4.1, the matrices that account for measurement information need to be kept separate of each other in order to trace back possible sources of double counting. When sensor node i receives an estimate from node j, it also has to keep and manage the matrix $\mathbf{S}^j_{k,\mathbf{R}^i}$ that is the corresponding matrix (3.14) from node j. The own and the received square root matrices are updated similarly to (3.13) by

$$\begin{aligned}\mathbf{S}^{\mathrm{f}}_{\mathbf{R}^i} &= \mathbf{F}^i \mathbf{S}^i_{\mathbf{R}^i} + \mathbf{F}^j \mathbf{S}^j_{\mathbf{R}^i}\,,\\ \mathbf{S}^{\mathrm{f}}_{\mathbf{R}^j} &= \mathbf{F}^i \mathbf{S}^i_{\mathbf{R}^j} + \mathbf{F}^j \mathbf{S}^j_{\mathbf{R}^j}\,.\end{aligned}$$

The bookkeeping of the received $\mathbf{S}^j_{\mathbf{R}^i}$ which accounts for measurements taken by sensor node j resembles (3.14), but differs because it is filled with zeros during further processing according to

$$\mathbf{S}^j_{k,\mathbf{R}^i} = \left[\mathbf{L}_k \mathbf{A}_k \mathbf{S}^j_{k-1,\mathbf{R}^i}\ ,\ \mathbf{0}\right], \tag{3.15}$$

to account for the fact that the measurement noise affecting node j is uncorrelated with the estimates at node i for the following time steps.

The square root matrix $\mathbf{S}^i_{\mathbf{R}^i}$ can be used in a later fusion step to reconstruct the cross-covariances stemming from the previous fusion step by

$$\mathbf{P}^{i,j}_{\mathbf{R}} = \mathbf{S}^i_{\mathbf{R}^i}(\mathbf{S}^j_{\mathbf{R}^i})^{\mathrm{T}} + \mathbf{S}^i_{\mathbf{R}^j}(\mathbf{S}^j_{\mathbf{R}^j})^{\mathrm{T}}\,, \tag{3.16}$$

where $\mathbf{S}^j_{\mathbf{R}^i}$ is the common information with node i that has been tracked in node j. Therefore, $\mathbf{S}^i_{\mathbf{R}^j}$ is the corresponding square root matrix to (3.15) that was generated by node j when it received information from i. The reconstructed cross-covariance matrix (3.16) has to be combined with $\mathbf{P}^{i,j}_{\mathbf{Q}} = \mathbf{S}^i_{\mathbf{Q}}(\mathbf{S}^i_{\mathbf{Q}})^{\mathrm{T}}$ representing the common process noise. Finally, this results in the full cross-covariance matrix

$$\mathbf{P}^{i,j} = \mathbf{P}^{i,j}_{\mathbf{Q}} + \mathbf{P}^{i,j}_{\mathbf{R}}\,,$$

thus, allowing the fusion of state estimates using (2.12) and (2.11).

This approach is applicable to extend the sample-based reconstruction of correlated estimation errors in decentralized network topologies. In this case, an additional sample set for every sensor node is needed to keep track of correlated measurement noise. However, this would result in a enormous amount of samples.

3.4.3 Conclusion to Fully Decentralized Track-Keeping of Cross-Covariances

The previous sections proposed an extension of the square root decomposition to keep track of correlated estimation errors in hierarchical and fully decentralized sensor networks. This extension can also be generalized to the sample-based reconstruction. For hierarchical fusion, the samples or square root matrices can be merged using the fusion gains, whereas, for decentralized sensor networks, additional track-keeping of measurement noise covariances is necessary. Furthermore, these square root matrices or sample sets accounting for double counting of measurement information cannot be merged but must stay separate for every sensor node. Therefore, the number of tracked components proliferates over time. However, this prohibitive in many applications with limited bandwidth.

3.5 Conclusions to Full Reconstruction of Cross-Covariances

This chapter introduced two methods for distributed track-keeping of correlation estimation errors. The first method uses deterministic samples to sample correlated noise covariances, while the second one calculates the square root decompositions of correlated noise covariances. Since the underlying principle of both methods is the same, they can be used interchangeably. However, the square root decomposition appears to be more intuitive and manageable while also requiring less data to be communicated.

Furthermore, this chapter proposed an extension of these methods to keep track of correlated estimation errors even when local estimators use different state-space representations or when correlations are hard to track due to complicated network topologies with information loops. However, the results show that special care has to be taken when separating a global system model into smaller subspaces that are estimated, as this can lead to inconsistent local estimators. A suitable approach to circumvent this problem is to execute the fusion in a sufficiently big, in the worst case the full, state space. When only a subspace of the state space should be fused, then dimensionality reduction can be executed before the communication. It can also be applied to the proposed track-keeping methods, allowing correct reconstruction of the cross-covariance. However, reducing dimensionality is essential to decrease computational costs, and further research is required to find suitable separation techniques to obtain lower-dimensional subspaces.

The extension to network topologies without a central fusion node shows that tracking correlated estimation errors using deterministic samples or square root decompositions of correlated noise covariances is also possible. When hierarchical fusion occurs, samples or square root matrices are merged using the fusion gain. However, this merging is impossible for square root matrices that account for the measurement information. The downside of these track-keeping methods is the growing amount of computational complexity and bandwidth requirements. In order to optimally reconstruct the cross-covariances, all correlated estimation errors need to be included in the additional information for fusion. Since a new correlated noise term is incorporated at every step, the tracked information's size grows linearly over time. This growth of additional data is prohibitive in many distributed estimation tasks, where computational power and bandwidth are limited. Moreover, old correlated estimation errors tend to approach zero because of the measurement update and are, therefore, less important for the reconstruction. This motivates the extension proposed in the following chapter to reduce the amount of tracked information.

CHAPTER
4

Partial Reconstruction of Cross-Covariances

Contents

4.1 Related Work 47
4.2 Partial Reconstruction of Cross-Covariances 48
4.2.1 Reducing the Amount of Tracked Information 49
4.2.2 Extension to Fully Decentralized Network Topologies 55
4.3 Conclusions to Partial Reconstruction of Cross-Covariances 62

As discussed in Chapter 1, sensor networks often have limited energy and communication bandwidth. Therefore, the limitation of communicated knowledge in networks is vital to the longevity and functionality of the sensor network. The previous chapter introduced two methods for keeping track of correlated estimation errors in distributed estimation using either deterministic samples or square root decompositions of correlated noise covariances. As seen in the last chapter, this additional knowledge about the correlated estimation errors is advantageous for the fusion since it obtains optimal fusion results. However, the bandwidth requirements grow over time because both methods need to keep track of every newly incorporated correlated noise covariance. Furthermore, old noise covariances converge to zero because of the measurement update. Therefore, they contribute less to the reconstruction of the cross-covariance matrix and produce a significant communication overhead.

The following chapter proposes the limitation of this additional knowledge by utilizing a sliding window approach. The subsequent extension results in a trade-off between additional knowledge and bandwidth requirements. Moreover, since decentralized sensor networks produce more correlated noise covariance that cannot be reduced easily, an extension is proposed for the fusion in hierarchical and decentralized sensor networks.

4.1 Related Work

In the following section, a sliding window approach is proposed. Therefore, correlated estimation errors are not fully but only partially tracked and reconstructed. This partial reconstruction leads to uncertain correlated estimation errors that need to be accounted for during the fusion step. Several approaches have been proposed to address the unknown dependencies between estimation errors. A common method is obtaining an upper bound for the correlation between estimation errors. This bounding technique yields suboptimal fusion results compared to using the actual cross-covariance for fusion. The most frequently used approach is Covariance Intersection

(CI), which was already mentioned several times in this thesis. CI proposes a family of upper bounds [67, 132]. Assuming two sensor nodes i and j are given, then the joint covariance matrix is constructed as

$$\mathbf{J} = \begin{bmatrix} \mathbf{P}^i & \mathbf{P}^{i,j} \\ \mathbf{P}^{j,i} & \mathbf{P}^j \end{bmatrix} \leq \begin{bmatrix} \frac{1}{\omega}\mathbf{P}^i & \mathbf{0} \\ \mathbf{0} & \frac{1}{1-\omega}\mathbf{P}^j \end{bmatrix} ,$$

where the weight ω is chosen to minimize the fused covariance matrix. CI is guaranteed to produce credible fusion results even when estimation errors are fully correlated and the authors in [111] show that CI is the optimal bounding algorithm. Many authors provide a detailed view on CI [26, 27], propose improved parameterization [2, 106, 113] or methods to further tighten the bound [1]. While CI makes no assumptions about the correlation and always yields credible results, it is often too conservative. The reason for this is that state estimates provided by the local estimators usually do not have fully correlated estimation errors, since the fusion is only beneficial if sensor nodes provide new and, therefore, uncorrelated information.

Therefore, several authors propose suboptimal fusion methods that make certain assumptions about the correlation between estimation errors. Often, it is assumed that estimates contain a correlated and an uncorrelated part. As we have seen from the previous chapter, it is possible to reconstruct correlated estimation errors. If the reconstruction is only carried out partially, there is still knowledge that can be exploited. It can be assumed that the joint covariance matrix can be split in a correlated and an uncorrelated part [69], which leads to a more generalized notation of CI also called Split Covariance Intersection (SCI). Even more in general, this can also be formulated as a joint covariance matrix, where one part has known correlation and the other has unknown correlation [133]

$$\mathbf{J} = \underbrace{\mathbf{J}^k}_{\text{known}} + \underbrace{\mathbf{J}^u}_{\text{unknown}} .$$

This separation is a practical approach to solving the problem of partial reconstruction. The known part of the joint covariance $\mathbf{J}^k$ corresponds to the covariances from the local estimators and the partially reconstructed cross-covariances, as previously proposed in Chapter 3. On the other hand, the unknown part $\mathbf{J}^u$ belongs to correlated estimation errors that are not tracked and therefore need to be bounded.

4.2 Partial Reconstruction of Cross-Covariances

This section is concerned with the partial reconstruction of cross-covariances, but only focuses on the track-keeping of correlated estimation errors using the square root decomposition of correlated noise covariances. The motivation for this limitation is that older entries in the square root matrix are easier to identify and discard, as discussed in Section 3.2.3. First, a sliding window approach is proposed for the fusion of state estimates in centralized network topologies with one dedicated fusion center, as proposed in [146]. Afterward, the proposed bounding technique is generalized to arbitrary network topologies that include hierarchical and decentralized fusion as proposed in [148, 149].

4.2.1 Reducing the Amount of Tracked Information

The following section proposes an approach to limit the correlated noise covariances contained in the square root matrix to a user-defined time horizon $\mathcal{T}_{\mathbf{Q}}$. The section begins by introducing a sliding window approach to limit the number of tracked components in the square root matrix to better understand the implications of incorrectly tracked correlated estimation errors. However, the disadvantage of this approach is that without accounting for all correlated estimation errors, the reconstructed cross-covariance is not big enough and leads to noncredible fused estimates. The results motivate the need for the bounding technique that is introduced in the following section.

A Discarding Correlated Estimation Errors

In the following section, only a limited number of noise covariances is kept in the square root matrix $\mathbf{S}^i_{\mathbf{Q}}$ of sensor node i. For this limitation, the square root matrix is divided into two parts

$$\mathbf{S}^i_{\mathbf{Q}} = \left[\mathbf{S}^i_{\mathcal{T}_{\mathbf{Q}}}, \mathbf{S}^i_{\mathbf{\Omega}_{\mathbf{Q}}}\right] ,$$

where $\mathbf{S}^i_{\mathcal{T}_{\mathbf{Q}}}$ is a moving horizon square root decomposition matrix

$$\mathbf{S}^i_{\mathcal{T}_{\mathbf{Q}}} = \left[\mathbf{\Sigma}^i_{k-\mathcal{T}+1,\mathbf{Q}}, \mathbf{\Sigma}^i_{k-\mathcal{T}+2,\mathbf{Q}}, \dots, \mathbf{\Sigma}^i_{k,\mathbf{Q}}\right] ,$$

that includes only noise covariances referring to process noise and common prior information up to a limited time horizon $\mathcal{T} = \mathcal{T}_{\mathbf{Q}}$. The matrix $\mathbf{S}^i_{\mathbf{\Omega}_{\mathbf{Q}}}$ includes all the other noise covariances referring to process noise and common prior information. This exclusion of correlated noise covariances results in a cross-covariance term smaller than the actual cross-covariance matrix. Therefore, the uncertainty of the fused estimate is underestimated. The following example visualizes this problem by using error ellipses of covariances.

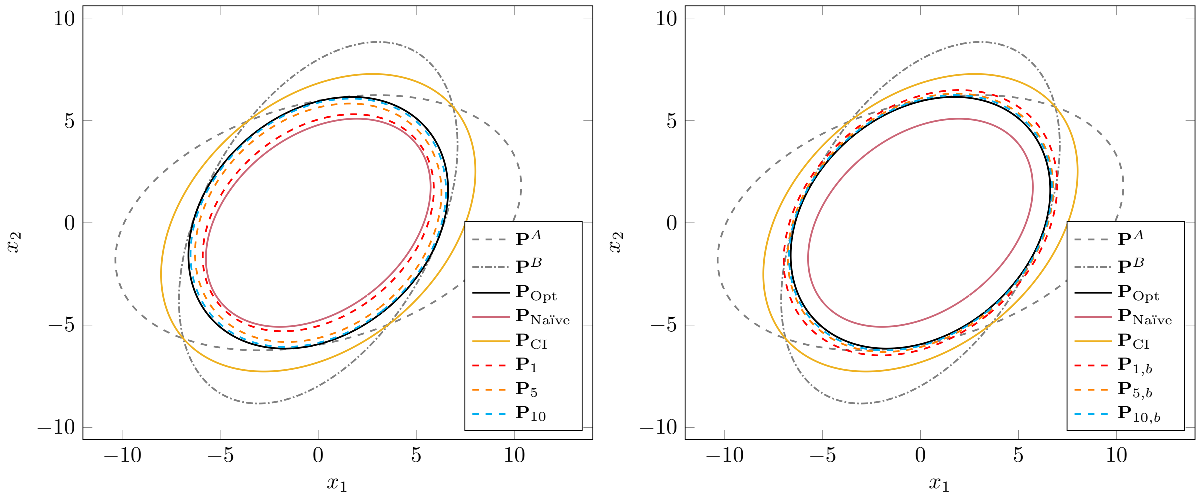
Example 2: Discarding Correlated Estimation Errors

The system from Example 1 is considered again. However, this time the square root decomposition (SqRD) is also used with three different parameterizations of the time horizon $\mathcal{T}_{\mathbf{Q}}$

- $\mathcal{T}_{\mathbf{Q}} = 1$, *only includes the last processing step,*
- $\mathcal{T}_{\mathbf{Q}} = 5$, *includes* 5 *processing steps, and*
- $\mathcal{T}_{\mathbf{Q}} = 10$, *includes all processing steps, but does not include the common prior information.*

Figure 4.1a depicts the error ellipses of the local covariance $\mathbf{P}^A$ *and* $\mathbf{P}^B$ *as well as the fused covariance matrix after* $k = 10$ *time steps.*

The remaining information not included in the square root matrix is discarded and not bounded. Figure 4.1a clearly shows, that the smaller the time horizon, the closer the error ellipse of the fused estimate is to the the naïve fusion result. Furthermore, the ellipses are always smaller than the error ellipse of optimal fusion, meaning that the fusion always underestimates the uncertainty.



(a) Fusion without bounding.

(b) Fusion with bounding.

Figure 4.1: Comparison of error ellipses for several fusion results using the square-root decomposition with and without bounding (adapted from [146]).

B Creating a Residual for Discarded Correlated Estimation Errors

To prevent the underestimation of uncertainty, an additional residual term $\boldsymbol{\Omega}_{\mathbf{Q}}$ is kept. It is recursively calculated and includes all noise terms $\mathbf{S}^i_{\boldsymbol{\Omega}_{\mathbf{Q}}}$ that are excluded from the matrix $\mathbf{S}^i_{\mathbf{Q}}$. At time step $k = 0$, the residual is initialized with

$$\mathbf{S}^i_{0,\boldsymbol{\Omega}_{\mathbf{Q}}} = \mathbf{0}\,.$$

Afterward, the square root matrix is processed as proposed previously and concatenated with the newest entries until the time horizon $\mathcal{T}_{\mathbf{Q}}$ is reached. When the time horizon is reached, the oldest noise terms from the square root decomposition matrix are excluded and added to the residual

$$\begin{aligned}\boldsymbol{\Omega}^i_{k,\mathbf{Q}} &= \boldsymbol{\Omega}^i_{k-1,\mathbf{Q}} + \boldsymbol{\Sigma}^i_{k-\mathcal{T},\mathbf{Q}}(\boldsymbol{\Sigma}^i_{k-\mathcal{T},\mathbf{Q}})^{\mathrm{T}} \\ &= \mathbf{S}^i_{\boldsymbol{\Omega}_{\mathbf{Q}}}\left(\mathbf{S}^i_{\boldsymbol{\Omega}_{\mathbf{Q}}}\right)^{\mathrm{T}}. \end{aligned} \tag{4.1}$$

This exclusion of noise covariances from the square root matrix can be seen as a shifting operation, which can be described mathematically as multiplying the matrix $\mathbf{S}^i_{\mathcal{T}_{\mathbf{Q}}}$ with a shift matrix $\mathbf{U}$

$$\mathbf{S}^i_{k-1,\mathcal{T}_{\mathbf{Q}}} := \mathbf{S}^i_{k-1,\mathcal{T}_{\mathbf{Q}}}\mathbf{U}\,.$$

The matrix $\mathbf{U}$ that shifts all entries n'_w positions to the left is constructed by

$$\mathbf{U} = \begin{bmatrix} \mathbf{0}_{n'_x \times n'} \\ \mathbf{I}_{n' \times n'} \end{bmatrix},$$

$$(n'_x, n') = \begin{cases} n'_x = 0,\; n' = n_w \tau, & \text{if } \tau \leq \mathcal{T}_{\mathbf{Q}} \\ n'_x = n_x,\; n' = n_w \mathcal{T}_{\mathbf{Q}}, & \text{if } \tau > \mathcal{T}_{\mathbf{Q}} \end{cases},$$

where n_x denotes the dimension of the system state space, and n_w is the dimension of the process noise covariance matrix. This operation results in the cancellation of the first noise covariance in the matrix. A new correlated noise covariance can be concatenated (see equation (3.4)) by

$$\mathbf{S}^i_{k,\mathcal{T}_\mathbf{Q}} = \mathbf{L}^i_k \left[\mathbf{A}_k \mathbf{S}^i_{k-1,\mathcal{T}_\mathbf{Q}}, \boldsymbol{\Sigma}^i_{k,\mathbf{Q}}\right] .$$

During the local filtering, the residual is updated by the prediction step

$$\boldsymbol{\Omega}^i_{k|k-1,\mathbf{Q}} = \mathbf{A}_k \boldsymbol{\Omega}^i_{k-1|k-1,\mathbf{Q}} \mathbf{A}^\mathrm{T}_k$$

and during the measurement update

$$\boldsymbol{\Omega}^i_{k|k,\mathbf{Q}} = \mathbf{L}^i_k \boldsymbol{\Omega}^i_{k|k-1,\mathbf{Q}} (\mathbf{L}^i_k)^\mathrm{T} .$$

This concludes the recursive calculation of the residual. This residual now requires a bounding method that constructs a tight bound.

C Partial Bounding of Discarded Correlated Estimation Errors

The previous section proposed to keep track of correlated estimation errors by using a sliding window approach. Therefore, the cross-covariance matrix $\mathbf{P}^{i,j}_\mathcal{T}$ includes all correlated noise covariances until the user-defined time horizon $\mathcal{T} = \mathcal{T}_\mathbf{Q}$. Furthermore, a residual $\boldsymbol{\Omega}_\mathbf{Q}$ is obtained to bound the remaining correlated estimation errors. In the following section, a fusion rule is formulated that can use the partially reconstructed cross-covariance and the obtained residual to obtain credible fusion results.

First, the optimal joint covariance matrix is considered

$$\mathbf{J} = \begin{bmatrix} \mathbf{P}^i & \mathbf{P}^{i,j} \\ \mathbf{P}^{j,i} & \mathbf{P}^j \end{bmatrix} .$$

However, the cross-covariance $\mathbf{P}^{i,j}$ is now only partially recovered. Therefore, the joint covariance matrix is formulated as follows

$$\mathbf{J} = \begin{bmatrix} \mathbf{P}^i & \mathbf{P}^{i,j}_\mathcal{T} + \mathbf{P}^{i,j}_{\boldsymbol{\Omega}} \\ \mathbf{P}^{j,i}_\mathcal{T} + \mathbf{P}^{j,i}_{\boldsymbol{\Omega}} & \mathbf{P}^j \end{bmatrix} ,$$

where the cross-covariances $\mathbf{P}^{i,j}$ is divided into two parts. The first part $\mathbf{P}^{i,j}_\mathcal{T}$ is reconstructed using

$$\mathbf{P}^{i,j}_\mathcal{T} = \mathbf{S}^i_{\mathcal{T}_\mathbf{Q}} (\mathbf{S}^j_{\mathcal{T}_\mathbf{Q}})^\mathrm{T} ,$$

while the second part $\mathbf{P}^{i,j}_{\boldsymbol{\Omega}}$ denotes the correlated noise covariances that are not kept track of and are, therefore, unknown.

The joint covariance matrix can be decomposed further by

$$\mathbf{J} = \begin{bmatrix} \mathbf{P}^i & \mathbf{P}^{i,j}_\mathcal{T} \\ \mathbf{P}^{j,i}_\mathcal{T} & \mathbf{P}^j \end{bmatrix} - \begin{bmatrix} \mathbf{P}^i_{\boldsymbol{\Omega}} & 0 \\ 0 & \mathbf{P}^j_{\boldsymbol{\Omega}} \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{P}^i_{\boldsymbol{\Omega}} & \mathbf{P}^{i,j}_{\boldsymbol{\Omega}} \\ \mathbf{P}^{j,i}_{\boldsymbol{\Omega}} & \mathbf{P}^j_{\boldsymbol{\Omega}} \end{bmatrix}}_{\mathbf{J}_{\boldsymbol{\Omega}}} .$$

Looking at (4.1), it can be found that

$$\mathbf{P}^i_{\boldsymbol{\Omega}} = \mathbf{S}^i_{\boldsymbol{\Omega}_{\mathbf{Q}}} \big(\mathbf{S}^i_{\boldsymbol{\Omega}_{\mathbf{Q}}}\big)^{\mathrm{T}} = \boldsymbol{\Omega}_{\mathbf{Q}} \,,$$

which is identical to the previously obtained residual $\boldsymbol{\Omega}^i_{\mathbf{Q}}$.

The entries on the block main diagonal of $\mathbf{J}_{\boldsymbol{\Omega}_{\mathbf{Q}}}$ are obtained by using the covariances from the local estimators. Sine the cross-covariances on the off-diagonals can not be reconstructed anymore, the aim is to find a bound for the correlations according to

$$\begin{bmatrix} \frac{1}{\omega_i}\boldsymbol{\Omega}^i_{\mathbf{Q}} & 0 \\ 0 & \frac{1}{\omega_j}\boldsymbol{\Omega}^j_{\mathbf{Q}} \end{bmatrix} \geq \begin{bmatrix} \boldsymbol{\Omega}^i_{\mathbf{Q}} & \mathbf{P}^{i,j}_{\boldsymbol{\Omega}} \\ \mathbf{P}^{j,i}_{\boldsymbol{\Omega}} & \boldsymbol{\Omega}^j_{\mathbf{Q}} \end{bmatrix} .$$

However, it is only possible to find a bound if the residual $\mathbf{J}_{\boldsymbol{\Omega}_{\mathbf{Q}}}$ is a valid cross-covariance matrix. The matrix $\mathbf{J}_{\boldsymbol{\Omega}_{\mathbf{Q}}}$ can be calculated by

$$\mathbf{J}_{\boldsymbol{\Omega}} = \begin{bmatrix} \mathbf{S}^i_{\boldsymbol{\Omega}_{\mathbf{Q}}} \\ \mathbf{S}^j_{\boldsymbol{\Omega}_{\mathbf{Q}}} \end{bmatrix} \begin{bmatrix} \mathbf{S}^i_{\boldsymbol{\Omega}_{\mathbf{Q}}} \\ \mathbf{S}^j_{\boldsymbol{\Omega}_{\mathbf{Q}}} \end{bmatrix}^{\mathrm{T}} ,$$

where $\mathbf{A}(\mathbf{A})^{\mathrm{T}} \geq \mathbf{0}$. It follows that $\mathbf{J}_{\boldsymbol{\Omega}_{\mathbf{Q}}}$ is a valid cross-covariance matrix and can be bounded. Finally, the bounded joint covariance matrix is defined by

$$\widetilde{\mathbf{J}} = \begin{bmatrix} \mathbf{P}^i - \boldsymbol{\Omega}^i_{\mathbf{Q}} & \mathbf{P}^{i,j}_{\mathcal{T}_{\mathbf{Q}}} \\ \mathbf{P}^{j,i}_{\mathcal{T}_{\mathbf{Q}}} & \mathbf{P}^j - \boldsymbol{\Omega}^j_{\mathbf{Q}} \end{bmatrix} + \begin{bmatrix} \frac{1}{\omega_i}\boldsymbol{\Omega}^i_{\mathbf{Q}} & 0 \\ 0 & \frac{1}{\omega_j}\boldsymbol{\Omega}^j_{\mathbf{Q}} \end{bmatrix} . \tag{4.2}$$

This joint covariance matrix can also be formulated for an arbitrary number of L sensor nodes

$$\widetilde{\mathbf{J}} = \begin{bmatrix} \mathbf{P}^1 - \boldsymbol{\Omega}^1_{\mathbf{Q}} & \mathbf{P}^{1,2}_{\mathcal{T}} & \dots & \mathbf{P}^{1,L}_{\mathcal{T}} \\ \mathbf{P}^{2,1}_{\mathcal{T}} & \mathbf{P}^2 - \boldsymbol{\Omega}^2_{\mathbf{Q}} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}^{L,1}_{\mathcal{T}} & \dots & \dots & \mathbf{P}^L - \boldsymbol{\Omega}^L_{\mathbf{Q}} \end{bmatrix} + \begin{bmatrix} \frac{1}{\omega_1}\boldsymbol{\Omega}^1_{\mathbf{Q}} & 0 & \dots & 0 \\ 0 & \frac{1}{\omega_2}\boldsymbol{\Omega}^2_{\mathbf{Q}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\omega_L}\boldsymbol{\Omega}^L_{\mathbf{Q}} \end{bmatrix} .$$

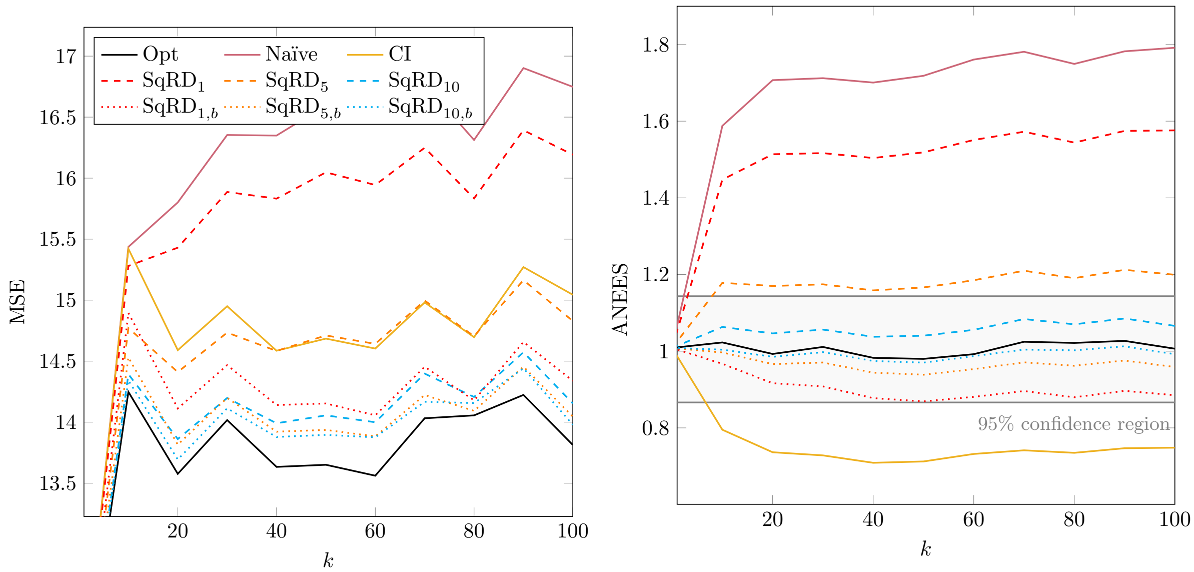
The weighting factor ω can be found by minimizing the fused covariance matrix according to formula (2.11). Alternatively, an approximate solutions such as the one proposed by [96, 110] can be used. While this approximation is suboptimal, it is simple to implement and has fast execution time. The weighting factor can be calculated by

$$\omega_i = \frac{1/\operatorname{tr}(\boldsymbol{\Omega}^i_{\mathbf{Q}})}{1/\operatorname{tr}(\boldsymbol{\Omega}^i_{\mathbf{Q}}) + 1/\operatorname{tr}(\boldsymbol{\Omega}^j_{\mathbf{Q}})} .$$

Then the fusion gain (see (2.10)) is

$$\mathbf{F}^j = \big(\mathbf{P}^i + \frac{1-\omega_i}{\omega_i}\boldsymbol{\Omega}^i_{\mathbf{Q}} - \mathbf{P}^{i,j}_{\mathcal{T}_{\mathbf{Q}}}\big)\big(\mathbf{P}^i + \frac{1-\omega_i}{\omega_i}\boldsymbol{\Omega}^i_{\mathbf{Q}} + \mathbf{P}^j + \frac{-\omega_i}{1-\omega_i}\boldsymbol{\Omega}^j_{\mathbf{Q}} - \mathbf{P}^{i,j}_{\mathcal{T}_{\mathbf{Q}}} - \mathbf{P}^{j,i}_{\mathcal{T}_{\mathbf{Q}}}\big)^{-1} .$$

The fused covariance and fused state can be calculated according to equations (2.11) and (2.12).



(a) MSE from 1000 test runs.

(b) Averaged Normalized Estimation Error Squared (ANEES) from 1000 test runs.

Figure 4.2: Comparison of CI, naïve fusion, optimal fusion using an infinite time horizon (Opt), square root decomposition-based fusion with time horizon $\mathcal{T}$ without bounding (SqRD$_{\mathcal{T}}$) and with bounding (SqRD$_{\mathcal{T},b}$) for the linear example [149].

Example 3: Bounding Discarded Correlated Estimation Errors

The same system as discussed in Example 2 is assumed. Futher, the proposed SqRD with time horizon of $\mathcal{T}_{\mathbf{Q}} = 1$ (only includes the last processing step), $\mathcal{T}_{\mathbf{Q}} = 5$ (includes 5 processing steps), and $\mathcal{T}_{\mathbf{Q}} = 10$ (includes all processing steps, but does not include the common prior information) is used. This time, the correlated noise covariances that are not included in the square root matrix, are collected in a residual term and bounded.

The error ellipses of the fused covariances using the proposed bounding method are depicted in Figure 4.1b. It can be observed that the smaller the time horizon, the closer the error ellipse of the fused estimate is to the fusion result using CI. Furthermore, the error ellipses of the fusion with residual are always larger than the error ellipse of the optimal fusion but significantly smaller than the fusion result of CI. Summarizing, these results indicate that the utilization of the tracked correlated noise covariances is beneficial to the fusion result and that the bounding of residual terms ensures the credibility of the fusion result.

D Evaluation

The following evaluation example was taken from [148] and uses the system description as discussed in Examples 1 and 2. Several fusion methods are evaluated to highlight the performance of the proposed SqRD method using different parameterizations of the time horizon $\mathcal{T}_{\mathbf{Q}}$. The fusion results are compared with the naïve fusion, which ignores the correlations between the sensor nodes and CI. Furthermore, the fusion result is also compared with the optimally reconstructed cross-covariance (see Section 3.2.1), where time horizon $\mathcal{T}_{\mathbf{Q}}$ is equal to the fusion step size plus

Method Abbreviation	Parameterization	Number of Additional Data Packages		
		$\mathbf{S}^i_{\mathbf{Q}}$ $(n_x^2 \times \mathcal{T}_Q)$	$\boldsymbol{\Omega}^i$ (n_x^2)	Sum
SqRD_1	$\mathcal{T}_Q = 1$	$4 \times 1 = 4$	0	4
$\mathrm{SqRD}_{1,b}$	$\mathcal{T}_Q = 1$	$4 \times 1 = 4$	4	8
SqRD_5	$\mathcal{T}_Q = 5$	$4 \times 5 = 20$	0	20
$\mathrm{SqRD}_{5,b}$	$\mathcal{T}_Q = 5$	$4 \times 5 = 20$	4	24
SqRD_{10}	$\mathcal{T}_Q = 10$	$4 \times 10 = 40$	0	40
$\mathrm{SqRD}_{10,b}$	$\mathcal{T}_Q = 10$	$4 \times 10 = 40$	4	44
$\mathrm{SqRD}_{\mathrm{Opt}}$	$\mathcal{T}_Q = 11$	$4 \times 11 = 44$	0	44

Table 4.1: Number of additional data packages sent by a single node i that are necessary for the reconstruction using SqRD, for the evaluation example of centralized fusion with two sensor nodes, every data package accounts for one entry of a matrix.

one to incorporate all correlated noise covariances. Just like in Example 2, the proposed SqRD is used with a time horizon of $\mathcal{T}_{\mathbf{Q}} = 1$ (only includes the last processing step), $\mathcal{T}_{\mathbf{Q}} = 5$ (includes 5 processing steps), and $\mathcal{T}_{\mathbf{Q}} = 10$ (includes all processing steps, but does not include the common prior information). The evaluation compares the proposed SqRD with bounding of residual terms ($\mathrm{SqRD}_{\mathcal{T},b}$) and without ($\mathrm{SqRD}_{\mathcal{T}}$).

Figure 4.2a shows the Mean Squared Error (MSE) over 1000 Monte Carlo Runs (MCR)s. The optimal fusion result obtained with the proposed method, including all noise terms, is the method that performs the best, while the naïve fusion is the method that performs the worst. The results show that the SqRD performs better the larger the time horizon while discarding correlation information instead of bounding worsens the performance. The credibility of the fused methods is evaluated in Figure 4.2b using the ANEES. CI is the most conservative method with the lowest ANEES, while the naïve fusion is the method with the highest ANEES. The SqRD without bounding is located between the optimal fusion result and the naïve fusion result, meaning that the uncertainty is underestimated to some degree. However, the SqRD with bounding is located close to the optimal fusion result but lies between the optimal fusion result and the fusion result of CI. These results suggest that the fusion without bounding produces noncredible and the fusion with bounding produces credible fusion results.

Table 4.1 shows the additional information a single sensor node i has to send to the fusion center depending on the parameterization of SqRD. Each entry of a matrix is counted as a data package. The square root matrix $\mathbf{S}^i_{\mathbf{Q}}$ includes the Cholesky decomposition of the common prior information and the common process noise. Both noise covariances have the same dimension $n_x = n_w = 2$, where n_x is the dimension of the state and n_w is the dimension of the process noise. Therefore each entry of the square root matrix has the size $n_x^2 = 4$. The residual $\boldsymbol{\Omega}^i$ has the size $n_x^2 = 4$ as well. The table shows that the amount of data packages increases with the time horizon $\mathcal{T}_{\mathbf{Q}}$. For completion, the table also shows how much additional data is caused by using the SqRD for the optimal reconstruction ($\mathrm{SqRD}_{\mathrm{Opt}}$). While the number of data packages is identical to $\mathrm{SqRD}_{10,b}$, it produces better results and would be preferable. However, $\mathrm{SqRD}_{1,b}$ and $\mathrm{SqRD}_{5,b}$ require less additional data, while still producing credible fusion results.

4.2.2 Extension to Fully Decentralized Network Topologies

Reducing communication bandwidth is an especially challenging task in networks without a dedicated fusion center. In centralized network topologies, the additional knowledge to perform the reconstruction can be reset after the fusion by reinitialization. Then, the common prior information is identical to the fused estimate, and the track keeping of correlated noise covariances can start from the beginning. However, decentralized network topologies do not allow for this reinitialization because the information is further passed on after the fusion step and cannot be reset. Therefore, additionally tracked information, e.g., in the form of the square root matrix, has to be reduced but further communicated. Furthermore, information that is only partially known, e.g., in the form of the residual, must also be communicated further and altered to incorporate previous fusion steps.

The following section proposes the partial reconstruction and subsequent bounding of correlated estimation errors in sensor networks without a central processing unit. As before, the hierarchical fusion of state estimates is extended first, as it is also a basis for the correlation in fully decentralized sensor networks. Afterward, the partial track keeping of correlated measurement information is proposed. Since the decomposition of measurement information can cause a significant communication overhead, this section proposes to keep track of uncorrelated measurement information to exploit known independence. Finally, this section closes with two evaluation examples, showing that the proposed partial reconstruction is beneficial in decentralized sensor networks.

A Hierarchical Fusion

The optimal reconstruction of cross-covariances in sensor networks with a hierarchical structure has been introduced in Section 3.4.1. As discussed there, the local square root matrices of two sensor nodes i and j are linearly combined using the fusion gains $\mathbf{F}^i$ and $\mathbf{F}^j$. The resulting square root matrix then contains the appropriate square root decomposition for the correlated noise covariances of the fused estimate and can be used to reconstruct the cross-covariance matrix with any other sensor node. The local square root matrices of node i and j are shortened until a user-defined time horizon $\mathcal{T}_{\mathbf{Q}}$. As a result of this limitation of the square root matrix, two local residual terms $\mathbf{\Omega}^i_{\mathbf{Q}}$ and $\mathbf{\Omega}^j_{\mathbf{Q}}$ are calculated that contain the information that is not explicitly tracked for the reconstruction. During the fusion step, these residuals now have to be linearly combined using the fusion gains. Because the fusion step includes a bounding technique, the weight ω has to be included as well. Thus, the residual becomes

$$\begin{aligned}\mathbf{\Omega}^{\mathrm{f}}_{\mathbf{Q}} &= \frac{1}{\omega}\mathbf{F}^i\mathbf{\Omega}^i_{\mathbf{Q}}(\mathbf{F}^i)^{\mathrm{T}} + \frac{1}{1-\omega}\mathbf{F}^j\mathbf{\Omega}^j_{\mathbf{Q}}(\mathbf{F}^j)^{\mathrm{T}} \\ &\geq \mathbf{F}^i\mathbf{\Omega}^i_{\mathbf{Q}}(\mathbf{F}^i)^{\mathrm{T}} + \mathbf{F}^i\mathbf{\Omega}^{i,j}_{\mathbf{Q}}(\mathbf{F}^j)^{\mathrm{T}} + \mathbf{F}^j\mathbf{\Omega}^{j,i}_{\mathbf{Q}}(\mathbf{F}^i)^{\mathrm{T}} + \mathbf{F}^j\mathbf{\Omega}^j_{\mathbf{Q}}(\mathbf{F}^j)^{\mathrm{T}} \,.\end{aligned}$$

This is a bound, since any information about $\mathbf{\Omega}^{j,i}_{\mathbf{Q}}$ has been discarded. The derived fusion of square roots and their residual terms serves as a basis for the fusion in fully decentralized sensor networks.

B Decentralized Fusion

As discussed in Section 3.4.2, the reconstruction of cross-covariances in fully decentralized sensor networks requires the additional track keeping of correlated measurement noise covariances.

Therefore, an additional square root matrix of the measurement noise covariances has to be stored and processed. Since these square root matrices cannot be merged during the fusion step like those for common prior information and common process noise, they tend to proliferate fast. As a result, the reduction of tracked measurement information is vital in many applications.

Following the concept introduced before, the number of covariances included in the square root matrix $\mathbf{S}^i_{\mathbf{R}^i} = [\mathbf{S}_{\mathcal{T}_\mathbf{R}}, \mathbf{S}_{\mathbf{\Omega}_\mathbf{R}}]$ can be reduced to a user-defined time horizon $\mathcal{T}_\mathbf{R}$

$$\mathbf{S}^i_{\mathcal{T}_\mathbf{R}} = \left[\mathbf{\Sigma}^i_{k-\mathcal{T}+1,\mathbf{R}^i}, \mathbf{\Sigma}^i_{k-\mathcal{T}+2,\mathbf{R}^i}, \dots, \mathbf{\Sigma}^i_{k,\mathbf{R}^i}\right] .$$

This time horizon for correlated measurement information has to be identical to the time horizon $\mathcal{T}_\mathbf{Q}$ for common process noise and common prior information. According to the previous approach, the excluded correlated noise terms are collected in a residual covariance $\mathbf{\Omega}_\mathbf{R}$. When the fusion step is executed, the residuals are fused by calculating the linear combination using the fusion gains $\mathbf{\Omega}^i_\mathbf{Q}$ and $\mathbf{\Omega}^j_\mathbf{Q}$ and the weighting factor ω. Hence, the fused residual is calculated according to

$$\mathbf{\Omega}^\mathrm{f}_\mathbf{R} = \frac{1}{\omega}\mathbf{F}^i\mathbf{\Omega}^i_\mathbf{R}(\mathbf{F}^i)^\mathrm{T} + \frac{1}{1-\omega}\mathbf{F}^j\mathbf{\Omega}^j_\mathbf{R}(\mathbf{F}^j)^\mathrm{T} .$$

When the fusion step is executed using (4.2), the residuals accounting for the process noise and the common prior information $\mathbf{\Omega}_\mathbf{Q}$ and the residual for correlated measurement information $\mathbf{\Omega}^i_\mathbf{R}$ have to be combined

$$\mathbf{\Omega}^i = \mathbf{\Omega}^i_\mathbf{Q} + \mathbf{\Omega}^i_\mathbf{R} \tag{4.3}$$

to bound all correlated noise covariance that are not explicitly tracked. While the partial reconstruction of correlated measurement covariances reduces the bandwidth requirements significantly, it still requires a significant amount of additional information. Furthermore, as discussed before, local estimates are usually not fully correlated since the fusion is only beneficial when local nodes provide new and, therefore, uncorrelated measurement information. For that reason, the following section explores the track keeping of uncorrelated measurements.

C Keeping Track of Uncorrelated Measurements

The treatment of correlated estimation errors due to double counting of measurement information can be simplified. Instead of explicitly keeping track of correlated measurement information to reconstruct cross-covariances, uncorrelated measurements can be tracked. The motivation for this approach is that local state estimates usually contain measurements that are exclusively known to the local sensor nodes and have not been shared with other sensor nodes before. Therefore, these measurements are uncorrelated, which can be exploited during the fusion step. The local covariance matrix of node i is rewritten

$$\mathbf{P}^i = \mathbf{P}^i_{\mathbf{Q},\mathcal{T}} + \mathbf{P}^i_{\mathbf{Q},\mathbf{\Omega}} + \mathbf{P}^i_\mathbf{R} ,$$

where $\mathbf{P}^i_{\mathbf{Q},\mathcal{T}}$ accounts for the part of the local covariance matrix referring to common prior information and common process noise that can be reconstructed, while $\mathbf{P}^i_{\mathbf{Q},\mathbf{\Omega}}$ is referring to the part that is not tracked and thus is collected in the residual $\mathbf{\Omega}^i_\mathbf{Q}$. Furthermore, $\mathbf{P}^i_\mathbf{R}$ represents measurement information that is possibly correlated. This term referring to possibly correlated measurement information can further be separated into a correlated and an uncorrelated part

$$\mathbf{P}^i_{\mathbf{R}} = \underbrace{\mathbf{P}^{i,+}_{\mathbf{R}}}_{\text{correlated}} + \underbrace{\mathbf{P}^{i,-}_{\mathbf{R}}}_{\text{uncorrelated}} \quad .$$

Measurements that have not been shared with other sensor nodes can be assumed uncorrelated. Therefore, only the covariance $\mathbf{P}^{i,+}_{\mathbf{R}}$ that refers to possibly correlated measurements has to be bounded to ensure credible fusion results. While it is cumbersome to keep track of correlated noise covariances, it is easy to keep track of measurement information that has not been shared yet. The uncorrelated measurement noise residual $\mathbf{P}^{i,-}_{\mathbf{R}}$ can be calculated recursively. It is created at time step $k = 1$ when the first measurement is incorporated in the local estimate

$$\mathbf{P}^{i,-}_{k,\mathbf{R}} = \mathbf{K}^i_k \mathbf{R}^i_k (\mathbf{K}^i_k)^{\mathrm{T}} \,.$$

Afterward, it is updated during the prediction step and a new measurement noise covariance is added when the next measurement is incorporated

$$\mathbf{P}^{i,-}_{k,\mathbf{R}} = \mathbf{L}^i_k \mathbf{A}_k \mathbf{P}^{i,-}_{k-1,\mathbf{R}} (\mathbf{A}_k)^{\mathrm{T}} (\mathbf{L}^i_k)^{\mathrm{T}} + \mathbf{K}^i_k \mathbf{R}^i_k (\mathbf{K}^i_k)^{\mathrm{T}} \,.$$

To ensure that the assumption of uncorrelated measurement noise is correct, $\mathbf{P}^{i,-}_{\mathbf{R}}$ needs to be reset to $\mathbf{P}^{i,-}_{\mathbf{R}} = \mathbf{0}$ as soon as the local estimate of i is shared with other sensor nodes. Finally, the residual accounting for possibly correlated measurement information is calculated by

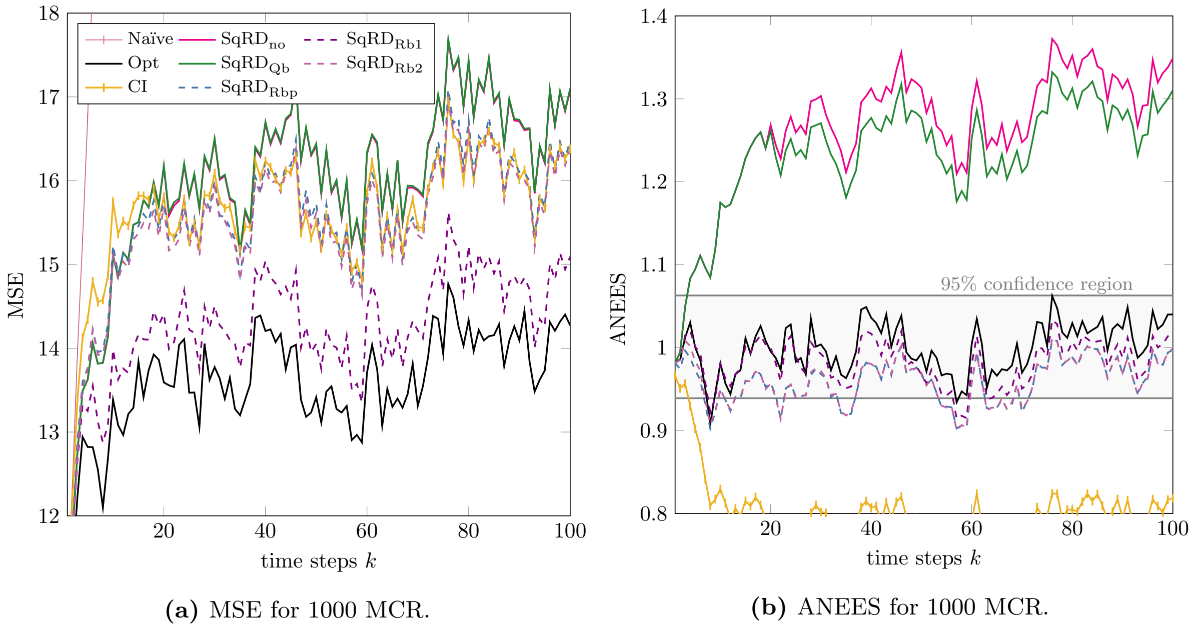
$$\mathbf{\Omega}^i_{\mathbf{R}} = \mathbf{P}^i - \mathbf{S}^i_{\mathcal{T}\mathbf{Q}} (\mathbf{S}^i_{\mathcal{T}\mathbf{Q}})^{\mathrm{T}} - \mathbf{\Omega}_{\mathbf{Q}} - \mathbf{P}^{i,-}_{\mathbf{R}} \,.$$

This residual $\mathbf{\Omega}^i_{\mathbf{R}}$ is combined with the residual for common prior information and common process noise (see (4.3)) and then used in the fusion rule (4.2). The proposed extension for the partial reconstruction of cross-covariances for the fusion in fully decentralized sensor networks is evaluated in the following section. First, a small example featuring two sensor nodes is evaluated. Afterward, a ring topology is assumed. Both evaluation examples are also featured in [148, 151].

D Evaluation of a Decentralized Example with Two Sensor Nodes

The first evaluation example is used to highlight the performance of the proposed partial reconstruction of cross-covariances using only two sensor nodes A and B that frequently exchange information. The system description is identical to the one in Examples 1 and 2. The data exchange between the two nodes is performed as follows

1. both sensor nodes execute a local filter update,
2. node A sends its local information to node B,
3. node B fuses information according to the selected fusion method and reinitializes its local state and covariance matrix with new fused information,
4. both sensor nodes execute a local filter update,
5. node B sends its local information to node A,
6. node A fuses information according to the selected fusion method and reinitializes its local state and covariance matrix with new fused information,
7. both nodes repeat the processing from the beginning.



(a) MSE for 1000 MCR.

(b) ANEES for 1000 MCR.

Figure 4.3: Comparison of the fusion results of different algorithms (adapted from [149]).

Figure 4.3 shows the MSE of different fusion methods after executing 1000 MCRs. The MSE of the naïve fusion, which neglects the cross-covariance completely, immediately diverges. The optimal track-keeping for a centralized fusion that reinitializes the local estimates after every fusion (Opt) shows the lowest MSE. The proposed square-root decomposition SqRD is shown in several different configurations to highlight how the bounding affects the fusion result. The time horizon for the square-root matrix accounting for common process noise and common prior information is chosen as $\mathcal{T}_{\mathbf{Q}} = 5$. The SqRD without bounding (SqRD$_{\text{no}}$) shows a relatively high MSE, because it does not account for older process noise covariance or any correlation due to measurement noise. The bounding of process noise (SqRD$_{\text{Qb}}$) performs slightly better, but also does not account for possibly correlated measurements. CI performs better than SqRD$_{\text{no}}$ and SqRD$_{\text{Qb}}$ but its performance is limited because it cannot exploit uncorrelated information. The proposed algorithm with partial bounding of measurement noise (SqRD$_{\text{Rbp}}$ (see Section 4.2.2.C)), shows better performance than CI. Finally, the SqRD using partial track-keeping of measurement information with time horizon $\mathcal{T}_{\mathbf{R}} = 5$ (SqRD$_{\text{Rb1}}$) and $\mathcal{T}_{\mathbf{R}} = 2$ (SqRD$_{\text{Rb2}}$) is evaluated. SqRD$_{\text{Rb1}}$ shows a lower MSE compared to all other methods, while the SqRD using a smaller time horizon SqRD$_{\text{Rb2}}$ is comparable to the performance of CI.

Figure 4.3 evaluates the credibility of the fused estimates using the ANEES. Fusion using the naïve approach diverges very fast and is therefore not included in the plot. CI on the other hand, is overly conservative and even far below the 95% confidence interval. The methods without bounding (SqRD$_{\text{no}}$ and SqRD$_{\text{Qb}}$) are noncredible since they do not appropriately approximate the uncertainty. The algorithm with partial bounding (SqRD$_{\text{Rbp}}$) is close to one, meaning that the actual MSE of the fused results matches the fused covariance matrix. Finally, the proposed methods using a limited time horizon to keep track of correlated measurement noise SqRD$_{\text{Rb1}}$ and SqRD$_{\text{Rb2}}$ are very close to the optimal fusion result but are slightly more conservative,

Method Abbreviation (and Reference)	Parameterization	Number of Additional Data Packages			
		$\mathbf{S}^i_{\mathbf{Q}}$ $(n_x^2 \times \mathcal{T}_Q)$	$\mathbf{S}^i_{\mathbf{R}^A}$, $\mathbf{S}^i_{\mathbf{R}^B}$ $(L \times n_x n_v \times \mathcal{T}_R)$	$\mathbf{\Omega}^i$ (n_x^2)	Sum
$\mathrm{SqRD}_{\mathrm{no}}$ (Sec. 4.2.1.A)	$\mathcal{T}_Q = 5$	$4 \times 5 = 20$	0	0	20
$\mathrm{SqRD}_{\mathrm{Qb}}$ (Sec. 4.2.1.C)	$\mathcal{T}_Q = 5$	$4 \times 5 = 20$	0	4	24
$\mathrm{SqRD}_{\mathrm{Rbp}}$ (Sec. 4.2.2.C)	$\mathcal{T}_Q = 5$	$4 \times 5 = 20$	0	4	24
$\mathrm{SqRD}_{\mathrm{Rb1}}$ (Sec. 4.2.2.B)	$\mathcal{T}_Q = 5$, $\mathcal{T}_R = 5$	$4 \times 2 = 20$	$2 \times 2 \times 5 = 20$	4	44
$\mathrm{SqRD}_{\mathrm{Rb2}}$ (Sec. 4.2.2.B)	$\mathcal{T}_Q = 5$, $\mathcal{T}_R = 2$	$4 \times 5 = 20$	$2 \times 2 \times 2 = 8$	4	32

Table 4.2: Number of additional data packages sent by a single node i that are necessary for the reconstruction using SqRD, for the evaluation example of decentralized fusion with two sensor nodes, every data package accounts for one entry of a matrix.

whereas $\mathrm{SqRD}_{\mathrm{Rb2}}$ shows similar performance to the proposed method with the partial bounding of correlated measurement errors ($\mathrm{SqRD}_{\mathrm{Rbp}}$).

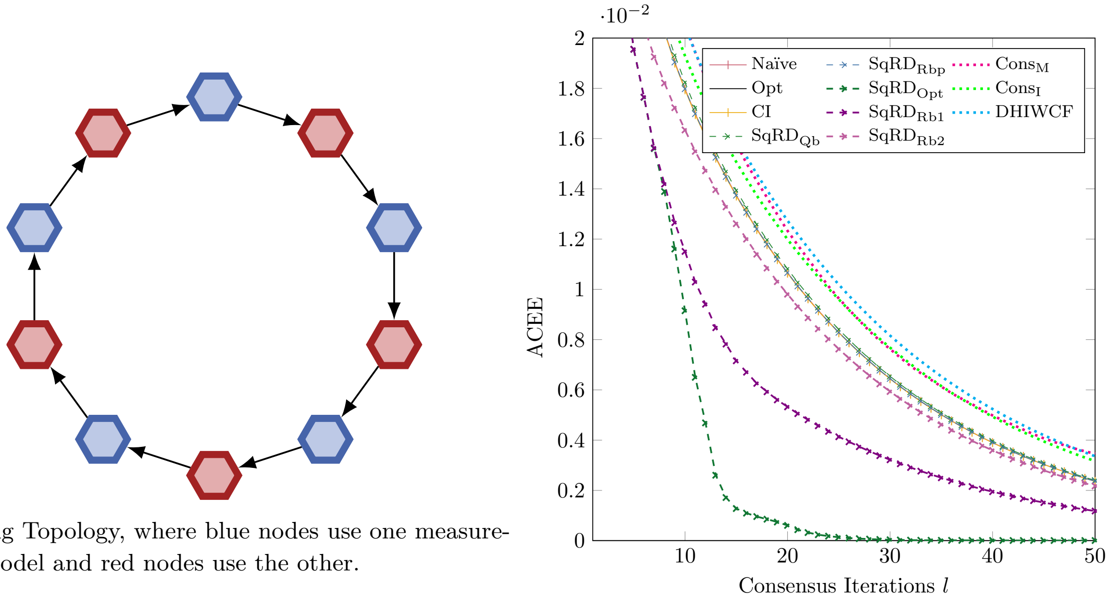
Table 4.2 shows the amount of additional data that every sensor node i has to process and communicate to the other node. As opposed to Table 4.1, the sensor nodes do not just process a square root matrix $\mathbf{S}^i_{\mathbf{Q}}$ and a residual $\mathbf{\Omega}^i$, but also a square root matrix for every sensor of the $L = 2$ nodes in the network. This results in two additional matrices $\mathbf{S}^i_{\mathbf{R}^A}$ and $\mathbf{S}^i_{\mathbf{R}^B}$. At the beginning of the processing, nodes A and B have not shared information. Therefore, before the nodes communicated the first time, node i only processes its matrix $\mathbf{S}^i_{\mathbf{R}^i}$. However, the nodes receive the corresponding matrix from the other node as soon as information is shared. Each additional entry to keep track of correlated measurement noise in $\mathbf{S}^i_{\mathbf{R}^A}$ and $\mathbf{S}^i_{\mathbf{R}^B}$ has the dimension $n_x n_v = 2$, where n_x is the dimension of the state and n_v is the dimension of the measurement noise. All residual terms that are produced are added into one residual before communication (see (4.3)). Therefore, the size of the residual is constant for all parameterizations of SqRD.

E Evaluating the Consensus between States

An important task in sensor networks is finding a global consensus between sensor nodes and it is predominantly solved by consensus methods. However, many consensus methods are only suboptimal fusion methods [31] and do not use knowledge about correlated estimation errors. The section shows, that the correlation of estimation errors is not just important for the credibility of the fused estimates, but it also speeds up convergence of the local state estimates towards a global consensus. To evaluate the convergence, the Averaged Consensus Estimate Error (ACEE), similarly defined in [87], is defined as

$$\mathrm{ACEE}(k) = \frac{1}{L} \sum_{i=1}^{L} \left(\hat{\underline{x}}^i_k - \bar{\underline{x}}_k \right) , \; \bar{\underline{x}}_k = \frac{1}{L} \sum_{i=1}^{L} \hat{\underline{x}}^i_k ,$$

where L is the number of sensor nodes and $\bar{\underline{x}}_k$ is the mean of all local estimates at time step k. This measure indicates the degree of consensus among estimates from all nodes in the network. Ideally, it should converge to zero, meaning that the state estimates in all sensor nodes are identical. A network of ten sensor nodes with ring topology (see Figure 4.4a) is considered. The system description is similar to the one in Example 2, but the measurement covariances are reduced to



(a) Ring Topology, where blue nodes use one measurement model and red nodes use the other.

(b) Convergence of state estimates towards a consensus.

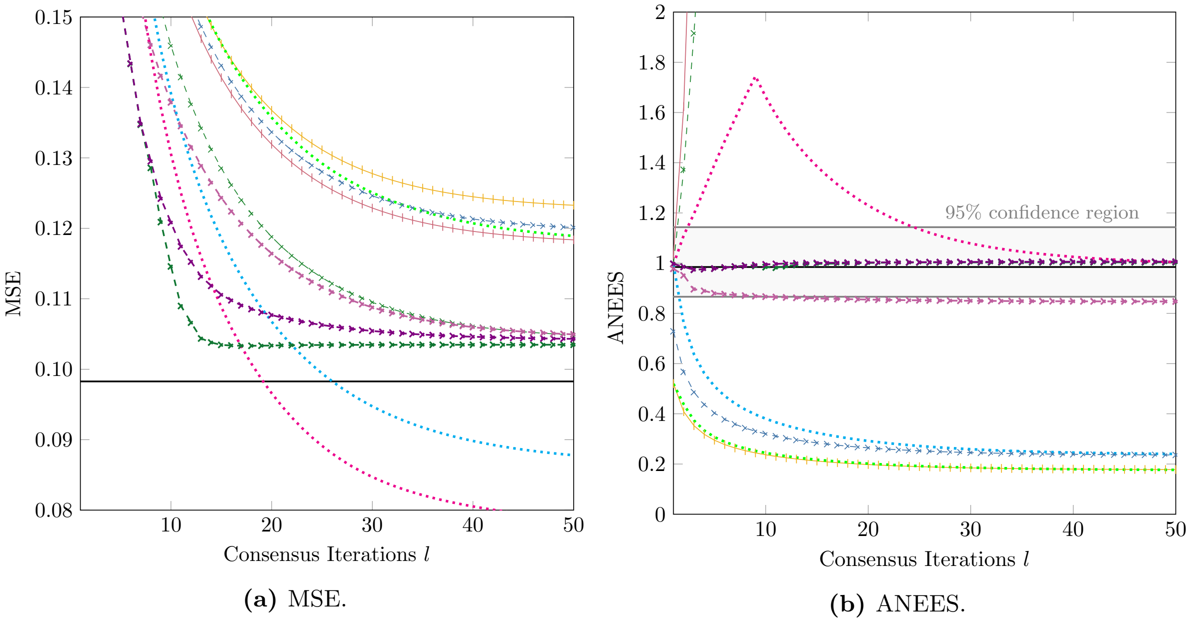
Figure 4.4: Visualization of a ring topology and comparison of consensus between estimates using different fusion methods (adapted from [149]).

$\mathbf{R}^A = 0.2$ and $\mathbf{R}^B = 0.02$, which is chosen for stability reasons. As can be seen in Figure 4.4a, the sensor nodes alternate between the measurement model of node A and node B. The sensor nodes first perform ten filtering steps independently and then communicate their local information to their neighbors multiple times.

For the evaluation, the fusion results of several consensus algorithms are compared as well, namely consensus on measurements [104] ($\mathrm{Cons_M}$), consensus on information [105] ($\mathrm{Cons_I}$) and hybrid consensus method called DHIWCF [87] that performs a consensus on measurement on the first iteration and a consensus on information afterwards. Consensus on information is performed using Metropolis weights. However, it should be noted that many consensus algorithms have been proposed in recent years and that the utilized algorithms may not be best tailored to the considered problem.

Figure 4.4b shows the convergence rate of the state estimates using the ACEE. CI and naïve fusion show very similar convergence rates. Surprisingly, all considered consensus methods converge slightly slower, while the hybrid consensus algorithm DHIWCF lies between consensus on measurements and consensus on information. Keeping track of all measurements ($\mathrm{SqRD_{Opt}}$), however, leads to the fastest convergence, followed by the SqRD with a time horizon $\mathcal{T}_{\mathbf{R}} = 3$ ($\mathrm{SqRD_{Rb1}}$), and using a time horizon $\mathcal{T}_{\mathbf{R}} = 1$. These results suggest that even a small time horizon for keeping track of correlated measurement noise causes significant performance difference. The time horizon of the square root matrix keeping track of the process noise is $\mathcal{T}_{\mathbf{Q}} = 11$, meaning that in this evaluation example, process noise and common prior information are fully tracked.

Figure 4.5a shows the MSE of the fused estimates averaged over all local estimates. The optimal track keeping of correlated noise covariances ($\mathrm{SqRD_{Opt}}$) achieves the lowest MSE of all fusion



(a) MSE.

(b) ANEES.

Figure 4.5: Evaluation results for fusion in ring topology performing several consensus steps (adapted from [149]).

methods, except the consensus methods, and almost approaches the result of the centralized optimal fusion result. SqRD with a smaller time horizon $\mathrm{SqRD}_{\mathrm{Rb1}}$ and $\mathrm{SqRD}_{\mathrm{Rb2}}$ performs well but converges more slowly. Consensus on Information ($\mathrm{Cons}_{\mathrm{I}}$) does not show any performance improvements in comparison to the other fusion methods. On the other hand, consensus on measurement converges slightly slower but outperforms all other methods after 22 time steps. The hybrid method DHIWCF shows slightly lower performance than consensus on measurements. Overall, consensus on measurement and DHIWCF reach a lower average MSE because the utilization of measurement information is more effective than the fusion of state estimates.

The credibility of the fused estimates is shown in Figure 4.5b, using the ANEES averaged over all sensor nodes. The ANEES is close to the optimal fusion result for $\mathrm{SqRD}_{\mathrm{Rb1}}$, $\mathrm{SqRD}_{\mathrm{Opt}}$, and $\mathrm{SqRD}_{\mathrm{Rb2}}$. Furthermore, the fusion method that bounds the correlated measurement noise partially ($\mathrm{SqRD}_{\mathrm{Rbp}}$) is close to CI, which are both relatively conservative and outside of the 95% confidence interval. Consensus on information ($\mathrm{Cons}_{\mathrm{I}}$) shows similar performance as CI but performs slightly worse because it was implemented using Metropolis weights that do not minimize the trace or the determinant. The performance of consensus on measurements ($\mathrm{Cons}_{\mathrm{M}}$) depends on the correction weights to mitigate the averaging of measurements [17]. Here, the correction weight is chosen as 2 in the first consensus step when only two measurements are available. Then, the correction weight is incremented by one in every consensus step until 10 to account for the ten measurements once a consensus is reached. Because of these averaging characteristics, the ANEES rises as some measurements have higher weights than others during the averaging, as this leads to double counting of measurements and therefore noncredibility. When the consensus is approached, the ANEES converges towards 1. Therefore, as soon as the consensus is fully reached, the method is optimal. DHIWCF shows slightly less conservative results than CI and reaches a relatively low

Method Abbreviation (and Reference)	Parameterization	Number of Additional Data Packages			
		$\mathbf{S}^i_{\mathbf{Q}}$ $(n_x^2 \times \mathcal{T}_Q)$	$\mathbf{S}^i_{\mathbf{R}^1}, \dots, \mathbf{S}^i_{\mathbf{R}^L}$ $(L \times n_x n_v \times \mathcal{T}_R)$	$\mathbf{\Omega}^i$ (n_x^2)	Sum
$\mathrm{SqRD}_{\mathrm{Opt}}$ (Sec. 3.4.2)	$\mathcal{T}_Q = 11$, $\mathcal{T}_R = 10$	$4 \times 11 = 44$	$10 \times 2 \times 10 = 200$	0	244
$\mathrm{SqRD}_{\mathrm{Rbp}}$ (Sec. 4.2.2.C)	$\mathcal{T}_Q = 11$	$4 \times 11 = 44$	0	4	48
$\mathrm{SqRD}_{\mathrm{Rb1}}$ (Sec. 4.2.2.B)	$\mathcal{T}_Q = 11$, $\mathcal{T}_R = 3$	$4 \times 11 = 44$	$10 \times 2 \times 3 = 60$	4	108
$\mathrm{SqRD}_{\mathrm{Rb2}}$ (Sec. 4.2.2.B)	$\mathcal{T}_Q = 11$, $\mathcal{T}_R = 1$	$4 \times 11 = 44$	$10 \times 2 \times 1 = 20$	4	68

Table 4.3: Number of additional data packages sent by a single node i that are necessary for the reconstruction using SqRD, for evaluation example of consensus in a hierarchical network, every data package accounts for one entry of a matrix.

MSE while still achieving credible results. However,the best trade-off between convergence rate, mean squared error, and credibility can be achieved using the proposed SqRD method.

Finally, the amount of additional data packages communicated by every sensor node i with its neighboring node is listed in Table 4.3. While $\mathrm{SqRD}_{\mathrm{Opt}}$ shows accurate and credible fusion results, it produces an enormous amount of additional data. On the other hand, the methods with a limited time horizon $\mathcal{T}_{\mathbf{R}}$ show very similar results while requiring fewer data. However, the track-keeping in decentralized sensor networks with many sensor nodes produces a lot of additional data.

4.3 Conclusions to Partial Reconstruction of Cross-Covariances

In this chapter, we proposed a sliding window approach to keep track of a limited amount of correlated estimation errors and partially reconstruct cross-covariances. By doing so, bandwidth requirements for the communication of additional knowledge are lowered. In addition, not explicitly tracked correlated noise covariances are collected in a residual term and bounded to ensure credible results. The resulting method can be tailored to design a trade-off between the credibility of the fusion result and the accuracy of the fused estimate. The evaluation examples also show that the track-keeping of correlated measurement noise is crucial to ensure fast convergence to a global consensus.

Unfortunately, the proposed framework still comes with a heavy burden on the communication bandwidth. However, many applications contain cyclic processes, e.g., the fusion step is executed every couple of time steps, and the local processing is relatively steady. Therefore, the proposed methods for track-keeping could be used to find patterns in the correlation of estimation errors during processing. Consequently, the full or partial reconstruction of cross-covariances could only serve as a tool to retrieve information and learn from it. The inferred partial knowledge about the cross-covariances can then be used, instead of the reconstruction methods, to fuse state estimates and obtain credible fusion results. Nevertheless, learning of correlated estimation errors can have pitfalls because of the special constraints cross-covariances or correlations have due to the semi-definiteness of the joint covariance matrix. Therefore, the next chapter investigates the learning of correlated estimation errors. Furthermore, an analytic approach for the learning of cross-correlations is proposed in Chapter 6 that relies on the proposed full or partial reconstruction methods.

CHAPTER
5

Learning Partial Knowledge about Correlation

Contents

5.1 The Cross-Correlation Matrix 63

5.2 Estimation of Uncertain Correlations 67

5.2.1 Sample Correlation 67

5.2.2 Bayesian Estimation of the Correlation Matrix 69

5.3 Conclusions to Learning Partial Knowledge about Correlation 73

The last two chapters proposed methods to fully or partially reconstruct cross-covariances for the fusion of state estimates from distributed sensor nodes. However, there are applications where the reconstruction is impossible, as no prior knowledge about the local estimators or the fusion is known. Furthermore, the sensor nodes might not be able to keep track of their estimation steps locally. When sensor systems are purchased and run third-party software, the implementation usually cannot be extended with additional track-keeping methods. Therefore, the proposed track-keeping methods of the previous chapters cannot be applied. Moreover, some sensor nodes might come with a tiny processing unit, and the local track-keeping using square roots or deterministic samples is computationally expensive. Therefore, the proposed methods might not run on small-scale systems with minimal computational power or memory storage.

This chapter proposes an approach for the estimation of correlations between estimation errors in distributed estimation tasks. For this estimation, the following chapter considers correlation matrices containing several correlation coefficients instead of the cross-covariance matrices since this is more intuitive and allows a deeper insight into the dependencies. The most pressing problem for the estimation task is that the correlation coefficients lie in a convex shape that results in a positive definite joint covariance matrix. Since the correlation coefficients cannot lie outside these natural bounds, their probability distribution also has to stay within them. First, this chapter discusses the cross-correlation matrix and its shape to understand the constraints the estimation task has to account for. Afterward, two methods for the estimation of correlated estimation errors are proposed and evaluated.

5.1 The Cross-Correlation Matrix

The previous section used cross-covariance matrices to measure the dependency between estimation errors. Using the cross-covariance for the fusion is beneficial since it can be calculated directly from known noise covariances and system parameters and construct the joint covariance matrix. However, the cross-covariance depends on the covariances of the local estimators. Therefore, it does not provide an intuitive assessment of the degree of dependency. On the other hand, the correlation matrix normalizes the cross-covariance matrix so that the local covariances do not have to be considered anymore. Therefore, the correlation matrix containing several correlation coefficients is a more intuitive way to measure the dependency between estimation errors.

The joint covariance matrix $\mathbf{J}$ of a p dimensional joint space is defined as [42]

$$\mathbf{J} = \begin{bmatrix} \sigma_1^2 & \dots & \sigma_{1p}^2 \\ \vdots & & \vdots \\ \sigma_{p1}^2 & \dots & \sigma_p^2 \end{bmatrix} ,$$

with cross-covariance $\sigma_{ij}^2 = \mathrm{E}\big[(\tilde{x}_i)(\tilde{x}_j)^{\mathrm{T}}\big]$ and variance $\sigma_i^2 = \mathrm{E}\big[(\tilde{x}_i)(\tilde{x}_i)^{\mathrm{T}}\big]$, where $\tilde{x}$ is the estimation error. Furthermore, the joint correlation matrix $\mathbf{X}$ is defined as

$$\mathbf{X} = \begin{bmatrix} 1 & \varrho_{12} & \dots & \varrho_{1p} \\ \varrho_{21} & 1 & & \vdots \\ \vdots & & \ddots & \vdots \\ \varrho_{p1} & \dots & \dots & 1 \end{bmatrix} , \tag{5.1}$$

which is the normalization of the joint covariance matrix $\mathbf{J}$. Both, the joint covariance matrix and the joint cross-correlation matrix are symmetric and positive semidefinite. Every correlation coefficient on the off-diagonals of the correlation matrix can be calculated by

$$\varrho_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_i \cdot \sigma_j}} .$$

Hence, every correlation coefficient has natural bounds $-1 \leq \varrho \leq 1$. While this normalization of the joint covariance matrix can be used as well for the estimation of correlation coefficients, another possible normalized version of the joint covariance matrix using the cross-correlation matrix $\mathbf{\Lambda}$ can be defined by

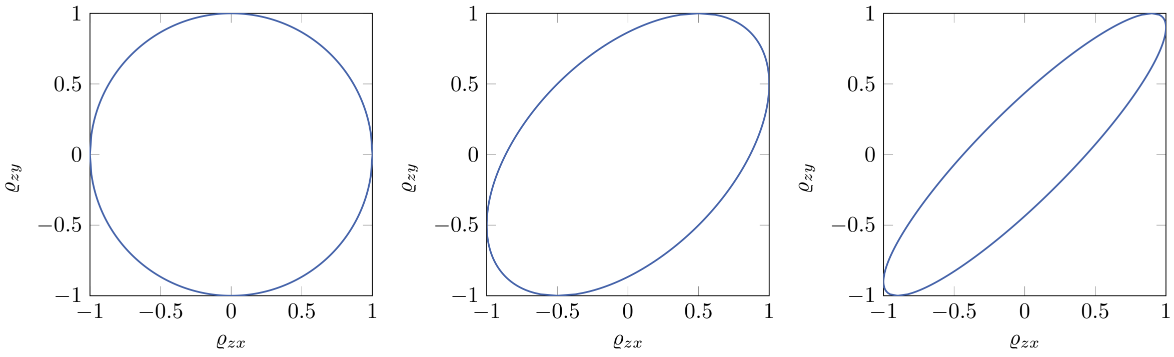
$$\mathbf{J} = \begin{bmatrix} \mathbf{S}^i & \mathbf{0} \\ \mathbf{0} & \mathbf{S}^j \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{\Lambda} \\ \mathbf{\Lambda}^{\mathrm{T}} & \mathbf{I} \end{bmatrix}}_{\mathbf{X}_{\mathbf{\Lambda}}} \begin{bmatrix} \mathbf{S}^i & \mathbf{0} \\ \mathbf{0} & \mathbf{S}^j \end{bmatrix}^{\mathrm{T}} , \tag{5.2}$$

where $\mathbf{S}^i$ and $\mathbf{S}^j$ are obtained by the Cholesky decompositions of the covariances $\mathbf{P}^i$ and $\mathbf{P}^j$. The correlation matrix $\mathbf{\Lambda}$ satisfies the natural bound

$$\mathbf{\Lambda}\mathbf{\Lambda}^{\mathrm{T}} \leq \mathbf{I} .$$

The advantage of the decomposition in (5.2), in comparison to the normalized joint covariance matrix (5.1), is that it results in a smaller set of correlation coefficients that need to be estimated. Furthermore, the cross-correlation included in the local covariances of the state estimates is not included in the correlation matrix $\mathbf{\Lambda}$. This is beneficial, since the correlations within the local covariances are given by the estimators and do not have to be learned.

The following section discusses, why the number of correlation coefficients that need to be estimated matters. While individual correlation coefficients can lie within their natural bounds $[-1, 1]$, the



(a) Circle with $\varrho_{yx} = 0.5$. **(b)** Ellipse with $\varrho_{yx} = 0.5$. **(c)** Ellipse with $\varrho_{yx} = 0.9$.

Figure 5.1: Convex Shapes of different correlation matrices in $\mathbb{R}^2$.

correlation matrix $\mathbf{X}$ has to be positive semi-definite to be a valid joint covariance matrix. Therefore, not all combinations of the correlation coefficients are possible. The interdependency between individual correlation coefficients and the parameterization of the correlation matrix is also discussed in [45, 64, 80, 117]. In order to properly learn and exploit correlations between estimation errors, the constraint of the correlation matrix have to be taken into account. In the simplest form, the correlation matrix with two correlation coefficients can be written as follows

$$\mathbf{X}_{\mathbf{\Lambda}} = \begin{bmatrix} \mathbf{I} & \mathbf{\Lambda} \\ \mathbf{\Lambda}^{\mathrm{T}} & 1 \end{bmatrix} = \left[\begin{array}{cc:c} 1 & 0 & \varrho_{zx} \\ 0 & 1 & \varrho_{zy} \\ \hdashline \varrho_{zx} & \varrho_{zy} & 1 \end{array}\right] .$$

In order for the correlation matrix $\mathbf{X}$ to be positive semidefinite, the determinant has to bigger or equal to zero, resulting in

$$\det(\mathbf{X}) = 1 - \varrho_{zx}^2 - \varrho_{zy}^2 \geq 0 ,$$

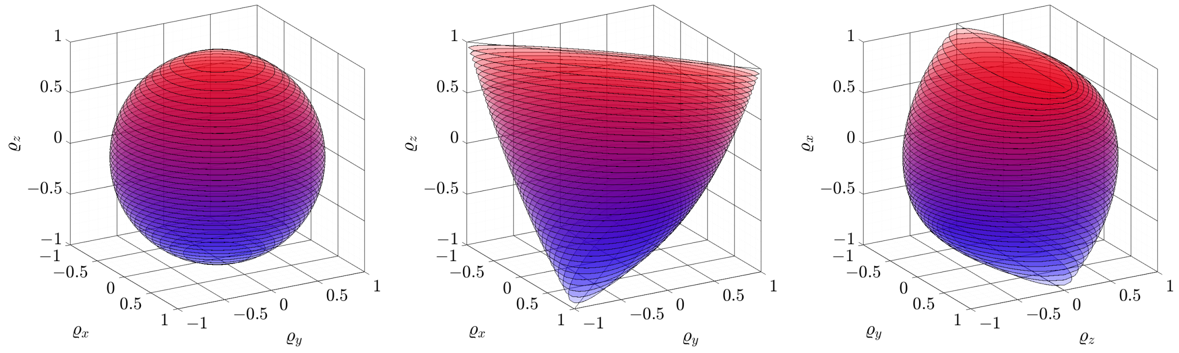
describing the shape of a circle that can be seen in Figure 5.1. However, if the normalized covariance matrix instead is defined by

$$\mathbf{X} = \left[\begin{array}{cc:c} 1 & \varrho_{yx} & \varrho_{zx} \\ \varrho_{yx} & 1 & \varrho_{zy} \\ \hdashline \varrho_{zx} & \varrho_{zy} & 1 \end{array}\right] ,$$

then the constraint for the correlation matrix changes to

$$\det(\mathbf{X}) = 1 - \varrho_{yx}^2 - \varrho_{zx}^2 - \varrho_{zy}^2 + 2\varrho_{yx}\varrho_{zx}\varrho_{zy} \geq 0 ,$$

describing an ellipse for which the correlation coefficient ϱ_{yx} determines how it is tilted. Figure 5.1 shows such an ellipse for $\varrho_{yx} = 0.5$ and $\varrho_{yx} = 0.9$. Therefore, by using the normalized joint covariance to calculate the correlation matrix, an additional correlation coefficient ϱ_{yx} is introduced because of the local covariance matrix. Then, this local covariance influences the constraints of the two correlation coefficients that are estimated. By using the correlation matrix $\mathbf{X}_{\mathbf{\Lambda}}$ instead, the influence of this local covariance matrix is excluded and the estimation is subject to less constraints that need to be taken care of.



(a) Sphere. **(b)** Elliptical tetrahedron. **(c)** Elliptical octahedron.

Figure 5.2: Convex Shapes of different correlation matrices in $\mathbb{R}^3$.

It is also possible to plot this convex shape of the correlation matrix depending on all three correlation coefficients. The result, called elliptical tetrahedron in [117], can be seen in Figure 5.2b. The plot shows the convex shape for $1 \leq \varrho_{yx} \leq 1$, where every ϱ_{yx} results in a different ellipse. Plotting the shape of different correlation matrices can be complicated because the number of components grows very fast, making it hard to plot them. However, there are two more correlation matrices that can be visualized in this context. The first one is

$$\mathbf{X} = \begin{bmatrix} \mathbf{I} & \mathbf{\Lambda} \\ \mathbf{\Lambda}^{\mathrm{T}} & 1 \end{bmatrix} = \left[\begin{array}{ccc|c} 1 & 0 & 0 & \varrho_x \\ 0 & 1 & 0 & \varrho_y \\ 0 & 0 & 1 & \varrho_z \\ \hline \varrho_x & \varrho_y & \varrho_z & 1 \end{array}\right] ,$$

where the constraint is determined by

$$\det(\mathbf{X}) = 1 - \varrho_x^2 - \varrho_y^2 - \varrho_z^2 \geq 0 \,.$$

The resulting convex shape is a sphere in $\mathbb{R}^3$ and can be seen in Figure 5.2a. Finally, the last correlation matrix that is visualized is

$$\mathbf{X} = \begin{bmatrix} \mathbf{I} & \mathbf{\Lambda} \\ \mathbf{\Lambda}^{\mathrm{T}} & \mathbf{I} \end{bmatrix} = \left[\begin{array}{cc|cc} 1 & 0 & \varrho_x & 0 \\ 0 & 1 & \varrho_y & \varrho_z \\ \hline \varrho_x & \varrho_y & 1 & 0 \\ 0 & \varrho_z & 0 & 1 \end{array}\right] ,$$

which results in the plot in Figure 5.2c. This shape is comparable to the one in [117] which is referred to as an elliptical octahedron, though the authors use a different parameterization of the correlation matrix.

The gist of this detailed discussion of correlation matrices and their shapes is to show that they are rather complex and learning needs to be done to not violate these constraints. It also becomes apparent that the uncertainty about the correlation coefficients has to lie within these constraints as well and is, therefore, not Gaussian distributed. In order to improve the fusion of state estimates, correlation coefficients are needed that are smaller than these constraints. The resulting smaller subset can be used to bound possible cross-covariances.

5.2 Estimation of Uncertain Correlations

Section 2.2.2 discussed different approaches to retrieve information about correlated estimation errors in distributed estimation. The described analytic approach leads to the full or partial reconstruction of cross-covariances and can be conducted as proposed in Chapter 3. However, as discussed in the beginning of this chapter, not every system is able to support this analytic approach. Therefore, the other strategy that could be taken is to use a simulation-based approach, where several MCRs of the complete system including local estimation and the fusion step are executed. Then, the correlation of estimation errors can be estimated from the simulated state estimates and the simulated system state.

The following section proposes estimation approaches for correlated estimation errors. First, usage of the sample correlation coefficient is proposed, as published in [150]. Afterward, an estimation approach using a particle filter is proposed. The following discussion shows, that particles are suitable to incorporate the natural constraints of the correlation coefficients and converge faster towards the true values.

5.2.1 Sample Correlation

The sample correlation coefficient is a method to statistically measure the Pearson correlation coefficient based on a number of observations. Let $\varepsilon_i(n)$ denote the estimation error of the ν-th element of the state at the n-th MCR

$$\varepsilon_i(n) = \mathbf{e}_\nu^\mathrm{T} \left(\underline{x}_k(n) - \hat{\underline{x}}^i_{k|k}(n) \right) ,$$

where $\underline{x}$ is the realization of the system state and $\mathbf{e}_\nu$ is the ν-th column of the identity matrix $\mathbf{I}$ of the corresponding dimension. The element index n and the time index k are dropped in $\varepsilon_i(n)$ for convenience. Using the samples of the errors of the i-th and j-th state estimates, the sample variances or sample covariances are calculated by

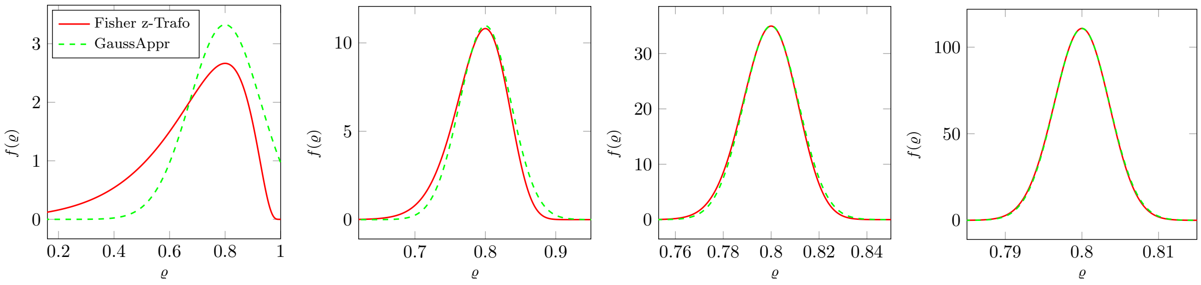
$$s_i = \tfrac{1}{N} \sum_{n=1}^{N} [\varepsilon_i(n)]^2 \quad , \quad s_j = \tfrac{1}{N} \sum_{n=1}^{N} [\varepsilon_j(n)]^2 \quad , \quad s_{i,j} = \tfrac{1}{N} \sum_{n=1}^{N} \varepsilon_i(n) \varepsilon_j(n) \, .$$

The sample variances and covariances then can be used to obtain the sample correlation coefficient

$$c^{i,j} = \frac{s_{i,j}}{\sqrt{s_i \, s_j}} .$$

It should be noted, that the sample correlation estimates the entries of the normalized joint covariance matrix $\mathbf{X}$.

Since the observations ε_i are noisy, the estimated sample correlation $c^{i,j}$ is uncertain. This uncertainty of the estimated correlation coefficient has to be accounted for during the fusion to ensure credible fusion results. However, the uncertainty of the sample correlation is rather difficult to calculate. The probability distribution of the correlation becomes extremely skewed towards the natural bounds of the correlation coefficients. Fisher [44], proposed to use a z-transformation



(a) $N = 10$ **(b)** $N = 100$ **(c)** $N = 1000$ **(d)** $N = 10000$

Figure 5.3: Comparison of the Fisher z-transform (red) with the simple Gaussian approximation of [19](green).

for a single sample correlation c, where $z = \frac{1}{2}\ln\left(\frac{1+c}{1-c}\right) = \operatorname{arctanh}(c)$. For the sample correlation c it approximately holds that

$$z \sim \mathcal{N}\Big(\frac{1}{2}\ln\Big(\frac{1+c}{1-c}\Big), \frac{1}{N-3}\Big) .$$

Therefore, the z-transformation can approximate the probability distribution of the sample correlation coefficient by a Gaussian distribution. This can be used by means of a confidence interval, e.g., $z \pm 1.96\frac{1}{\sqrt{N-3}}$. The inverse transformation is given by $c = \frac{e^{2z}-1}{e^{2z}+1} = \tanh(z)$. The limits of the confidence intervals can be treated as bounds of the correlation and used for the fusion as will be explained in Chapter 6. For large values of N, the probability distribution of the sample correlation coefficient is almost identical to a Gaussian ditribution [19] with the variance

$$\sigma_{i,j}^2 = \frac{\left(1-(c^{i,j})^2\right)^2}{N-2} . \tag{5.3}$$

This approximation is visualized for a different number of observations in Figure 5.3. While (5.3) is simpler for the specification of confidence intervals, it leads to a probability mass outside of the admissible area of the correlation coefficients. The Fisher z-transform does not suffer from this problem, but it leads to a probability distribution that is skewed to one direction. In both cases, using the Fisher z-transformation or the approximation from [19], only a single sample correlation coefficient is assumed, and no higher-dimensional approximations are given.

The sample correlation estimates only a single Pearson correlation coefficient. This is a significant downside, as it means that the sample correlation is agnostic to the other correlation coefficients and does not ensure a positive semidefinite joint covariance matrix. As discussed in Section 5.1, several correlation coefficients are dependent on each other to ensure a positive semidefinite joint covariance matrix. Therefore, the sample correlation estimation can result in estimates that violate the constraints of the correlation matrix $\mathbf{X}$, e.g., with a small number of observations or when the actual correlation is close to the edge of the constraint. Therefore, the following section proposes a method that is sensitive to the natural constraints of the correlation matrix. Furthermore, it is possible to freely choose the parameterization of the correlation matrix, using either the coefficients from the normalized joint covariance matrix $\mathbf{X}$ or the partial correlation matrix $\mathbf{X}_{\mathbf{\Lambda}}$. The proposed approach can also account for the skewness of the uncertainty distribution of the estimated correlation.

5.2.2 Bayesian Estimation of the Correlation Matrix

The estimation of the correlation coefficients is a nonlinear estimation task. Hence, a linear filter, e.g., the Kalman filter, cannot be used. Therefore, the problem of nonlinear filtering [116] can be formulated as follows. It is assumed that the parameterization of the joint correlation matrix is equal to (5.2) and that the correlation matrix $\mathbf{\Lambda}$ is static. Since it does not change over time, no model of the time evolution is necessary, and the prediction step of the filter is omitted. However, depending on the application, a state transition model to predict the correlation changes over time can be beneficial. The aim is to estimate a set of parameters $\underline{\lambda} = [\rho_1, \dots, \rho_{n_\varrho}]^{\mathrm{T}}$ of a correlation matrix $\mathbf{\Lambda}(\underline{\lambda})$, where n_ϱ is the number of the correlation coefficients that form the correlation space $\mathbb{R}^{n_\varrho}$. It is assumed that there is a prior probability density function (pdf) on the parameters $p_0(\underline{\lambda})$. Using Bayes rule, the aim is to estimate the posterior pdf $p(\underline{\lambda} \,|\, \mathbf{Y})$ of the parameter set given observations $\mathbf{Y} = [\underline{y}_1, \dots, \underline{y}_N]$ where N is the number of observations. Therefore the update of the prior pdf is calculated as

$$p(\underline{\lambda} \,|\, \mathbf{Y}) = \frac{\mathcal{L}(\mathbf{Y} \,|\, \underline{\lambda})\, p_0(\underline{\lambda})}{\int \mathcal{L}(\mathbf{Y} \,|\, \underline{\lambda})\, p_0(\underline{\lambda})\, d\underline{\lambda}}\,, \tag{5.4}$$

where $\mathcal{L}(\mathbf{Y} \,|\, \underline{\lambda})$ is the likelihood of the observation. This likelihood function has to be chosen so that correlation coefficients that best fit the observations have a higher likelihood than other correlation coefficients. The following section is concerned with the formulation of a likelihood function that fits this criteria.

A The Likelihood Function for the Correlation Matrix

In order to estimate the correlation, the conditional likelihood that a parameter vector $\underline{\lambda}$ describes the correlated estimation errors is calculated. Consequently, a likelihood function is used to evaluate whether the sampled correlations fit the observed data. To estimate the correlation, the estimation errors have to be calculated first. Then, these estimation errors serve as observations for which the likelihood of the sampled correlations is evaluated. The likelihood of a single observation of the joint state vector $\underline{\hat{m}}_n = \left[(\underline{\hat{x}}^i)^{\mathrm{T}}, (\underline{\hat{x}}^j)^{\mathrm{T}}\right]^{\mathrm{T}}$ with a Gaussian distribution is

$$\begin{aligned} f(\underline{\hat{m}}_n \,|\, \underline{\lambda}) &= \frac{1}{c} \exp\Big(-\frac{1}{2}(\underline{\hat{m}}_n - \mathbf{H}\underline{x})^{\mathrm{T}} \big(\mathbf{J}(\mathbf{\Lambda}(\underline{\lambda}))\big)^{-1} (\underline{\hat{m}}_n - \mathbf{H}\underline{x})\Big) \\ c &= (2\pi)^{N/2} \sqrt{\det(\mathbf{J}(\mathbf{\Lambda}(\underline{\lambda})))}\,, \end{aligned}$$

where $\mathbf{H}$ is the mapping matrix, which determines how the local state estimates are mapped into the global state estimate (see (2.9)), and c is the normalization constant. Furthermore, $\mathbf{J}(\mathbf{\Lambda}(\underline{\lambda}))$ denotes the joint covariance matrix of $\underline{\hat{m}}$, which is a function of the correlation matrix $\mathbf{\Lambda}(\underline{\lambda})$ that is characterized by a parameter vector $\underline{\lambda}$. The vector $\underline{x}$ is the realization of the state. Therefore, $\underline{\hat{m}}_n - \mathbf{H}\underline{x}$ denotes the estimation error of the n-th observation. The likelihood $\mathcal{L}$ of all N observations is

$$\mathcal{L}(\underline{\hat{m}}_1 \dots \underline{\hat{m}}_N \,|\, \underline{\lambda}) = f(\underline{\hat{m}}_1 \,|\, \underline{\lambda}) \dots f(\underline{\hat{m}}_N \,|\, \underline{\lambda})\,.$$

Therefore the likelihood for the Gaussian distribution is

$$\mathcal{L}(\underline{\hat{m}}_1 \ldots \underline{\hat{m}}_N \,|\, \underline{\lambda}) = \frac{1}{c} \exp\Big(- \frac{1}{2} \sum_{n=1}^{N} (\underline{\hat{m}}_n - \mathbf{H}\underline{x})^{\mathrm{T}} \big(\mathbf{J}(\mathbf{\Lambda}(\underline{\lambda}))\big)^{-1} (\underline{\hat{m}}_n - \mathbf{H}\underline{x}) \Big) \,.$$

It is also possible to use the log likelihood function instead, as it is monotonically related to the likelihood function. The joint covariance matrix can be constructed using the correlation matrix according to (5.2). Therefore, the estimation aims to find a parameter vector $\underline{\lambda}$ that has the highest likelihood.

The update in (5.4) using this likelihood function is unfortunately not easy to obtain, because it requires solving a complicated integral. A suitable solution are Monte Carlo integration methods since they are not subject to Gaussian constraints and have favorable convergence properties [40, 116].

B Sequential Importance Sampling for Correlation Estimation

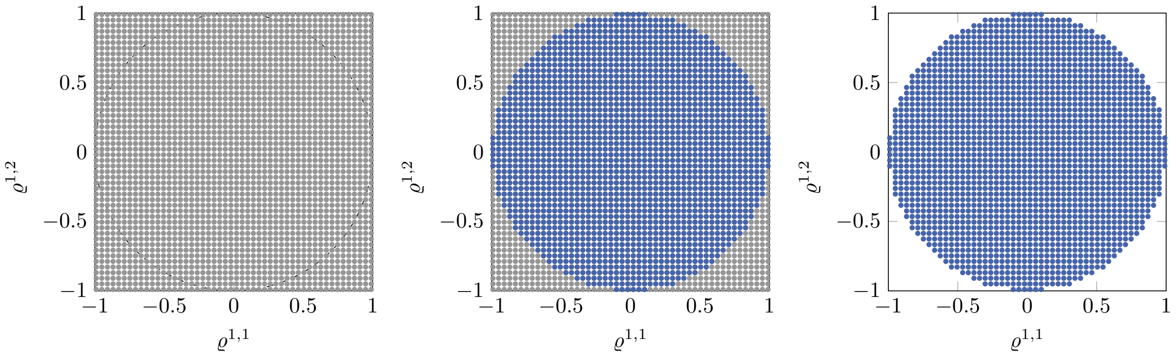
The estimation of a set of correlation coefficients is similar to the estimation of system parameters, for which several approaches have been published [89]. However, as discussed earlier, the analytic calculation of the correlation is not feasible, and therefore a Monte Carlo integration method is adopted. The idea is to use several particles that approximate the posterior pdf of the correlation coefficients based on several observations. Importance sampling is a popular example of such a method. However, it is not suitable for recursive estimation in its simplest form [40], which would mean that all observations have to be taken in advance and the estimation is done on the complete data. Therefore, Sequential Importance Sampling (SIS) is adopted.

First, a set of random samples $\{\underline{\lambda}\}_{m=1}^{M}$ consisting of M particles with weights $\{\omega\}_{m=1}^{M}$ subject to $\sum_{n=1}^{N} \omega_n = 1$ is drawn from a proposal distribution $\pi(\underline{\lambda})$, e.g., a uniform distribution. Then, the local estimators execute their individual filtering steps to obtain state estimates that are correlated, as discussed in Section 2.1.3. Afterward, the estimation errors are calculated using the state estimates and the realization of the true state of the system. Last, for every particle $\underline{\lambda}_m$ a joint covariance matrix $\mathbf{J}(\mathbf{\Lambda}(\underline{\lambda}_m))$ is created and used to evaluate the likelihood function. The likelihood function increases the weight of the particles that are more likely to originate from the correct joint distribution.

Since this reweighing of the particles leads to particle impoverishment, the effective sample size M_{eff} is used

$$M_{\text{eff}} = \frac{\left(\sum_{m=1}^{M} \omega_m\right)^2}{\sum_{m=1}^{M} (\omega_m)^2} \,.$$

If M_{eff} falls below a threshold $M_{\text{thr}} = M/2$, then importance resampling [121] is applied to resample the particle set. This resampling technique discards low-weight particles and resamples existing samples several times depending on their importance weight. However, to improve the sampling of the distribution, the particles have to be roughened [40, 54]. Therefore, Gaussian white noise with the variance $\sigma_{\text{jitter}} = \frac{KE}{N}$ is applied to every resampled particle, where E is the difference between the maximum and the minimum values of the particle component and K is a constant turning parameter (we used $K = 2$). When resampling, one must ensure that the roughened samples do



(a) Initially sampled grid.

(b) Using Constraints.

(c) New sample set.

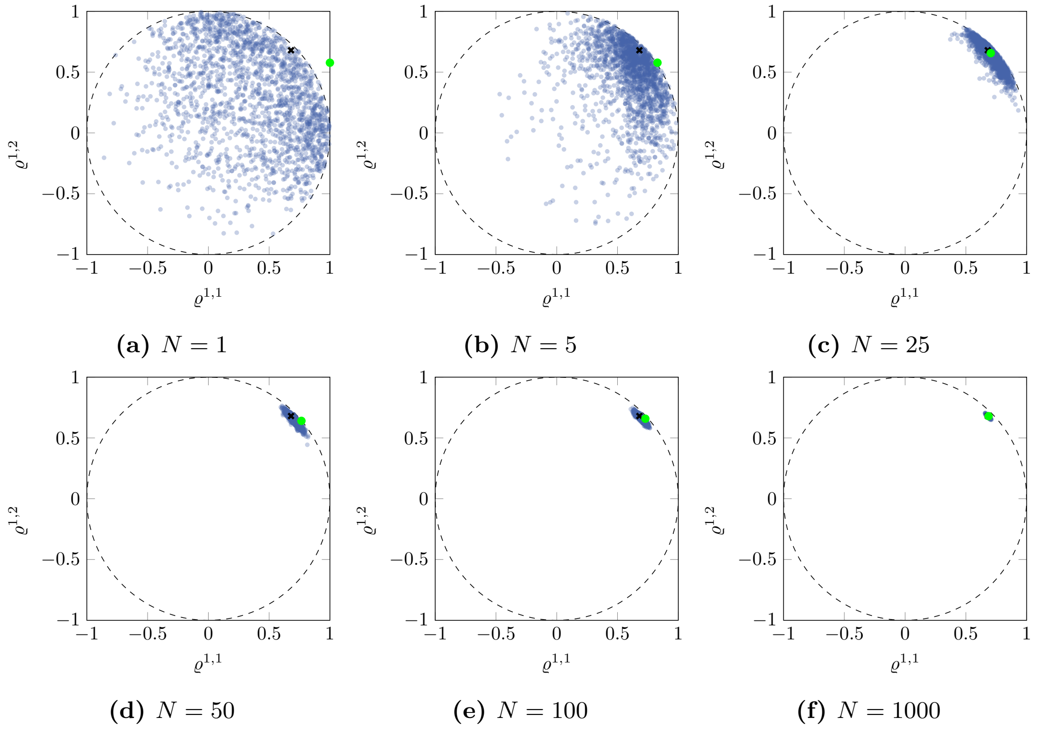
Figure 5.4: Example of Sampling a uniform distribution of a correlation matrix with two correlation coefficients, initial samples (gray) and samples that fit within the constraints (blue).

not violate the constraint for admissible correlation matrices. Therefore when roughening, the particles need to be checked if the correlation matrix is admissible. If the particle is not admissible, the roughening has to be removed and applied to the initial particle again until the particle is admissible.

C Prior Knowledge on the Correlation Matrix

As described in the previous section, the SIS approach requires the usage of a prior proposal density $\pi(\underline{\lambda}_0)$. In the case of estimating the correlation of estimation errors, there is usually no known prior on the correlation matrix $\mathbf{\Lambda}$ other than the natural constraint of the correlation coefficients. Therefore, a uniform distribution on all possible correlation coefficients is a suitable way to include prior knowledge. Because of the natural constraints of the correlation matrix, such a uniform distribution cannot easily be constructed. Therefore, a grid sampling method is proposed, where an initial set of particles containing M_0 number of particles in the $\mathbb{R}^{n_\varrho}$ space (see Figure 5.4) is generated. Afterward, the constraint on the correlation matrix is applied to reject all particles for which the joint covariance matrix is not positive semi-definite. Last, all particles are assigned the same weight $\omega_m = 1/M$, where M is the number of particles that fulfill the constraint.

This approach is straightforward and results in an almost uniform distribution. Nevertheless, depending on the constraint, it is unclear how many particles survive this rejection process, resulting in an odd number of particles. If a certain number of particles is needed, the sampling could be repeated with a different number of prior particles, or the user could calculate the correct number of necessary particles in advance. For the estimation of correlation coefficients, other priors than the uniform prior have been proposed [48]. Different priors could either be achieved by different sampling schemes or different weighting of the proposed particles, which could increase the weight for correlation coefficients that are known to be more probable. We tried using different priors with a one-dimensional correlation estimation problem but found that the uniform prior performed well in all experiments. Another possible sampling method could be to use Fibonacci grids [50, 128], which would result in a prior particle set with interesting geometrical properties. However, depending on the constraint, a suitable Fibonacci grid might be complicated to create.



(a) $N = 1$ **(b)** $N = 5$ **(c)** $N = 25$

(d) $N = 50$ **(e)** $N = 100$ **(f)** $N = 1000$

Figure 5.5: Particle distribution (blue) of the proposed filter at different number of observations N, true correlation coefficients (marked with black x) and sample correlation (green).

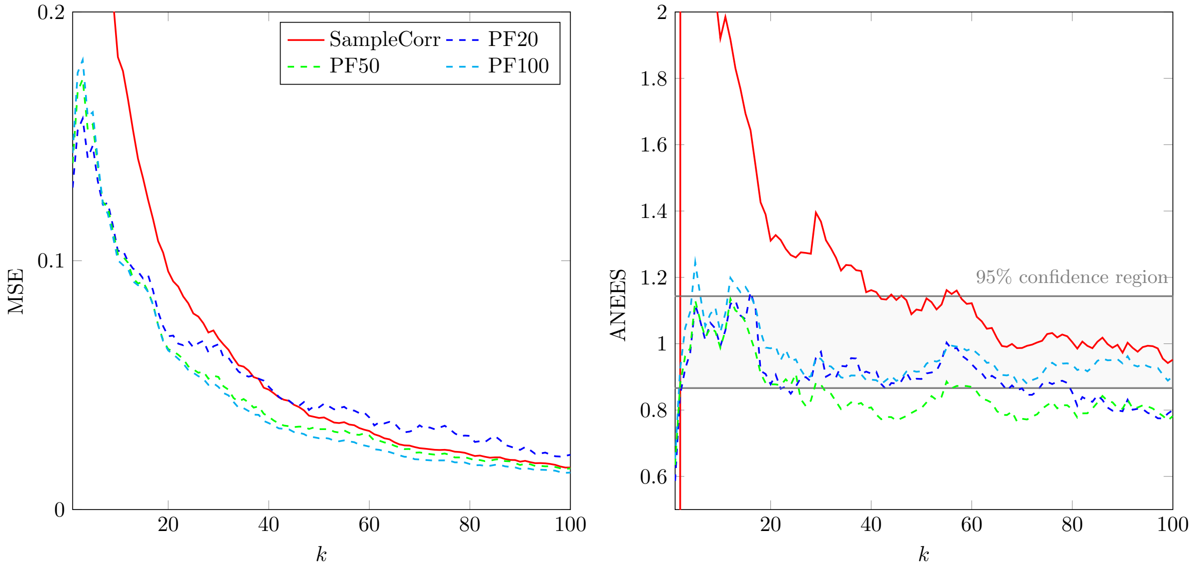
D Evaluation Estimating a two-dimensional Correlation Matrix

The following section evaluates the proposed methods to estimate the correlation of estimation errors from two local estimators. Let the joint correlation matrix be given by

$$\mathbf{X}_{\mathbf{\Lambda}} = \begin{bmatrix} \mathbf{I} & \mathbf{\Lambda} \\ \mathbf{\Lambda}^{\mathrm{T}} & 1 \end{bmatrix} ,$$

where the covariance matrices of the local trackers are $\mathbf{P}^A = [4, 1; 1, 1]$ and $\mathbf{P}^B = 1$, and the correlation matrix $\mathbf{\Lambda}$ fulfills the constraint $\mathbf{\Lambda}\mathbf{\Lambda}^{\mathrm{T}} \leq \mathbf{I}$. For this example, the correlation matrix $\mathbf{\Lambda}$ consists of two correlation coefficients ρ_1 and ρ_2, which lie in a circular shape in $\mathbb{R}^2$ as discussed in Section 5.1. The ground truth of the correlation matrix is $\mathbf{\Lambda} = [0.68, 0.68]^{\mathrm{T}}$, which is close to the outer edge of the natural constraint.

Figure 5.5 shows the estimated point cloud at a different number of observations. The point cloud of the particle filter converges towards the true value of the correlation coefficients. Furthermore, the particle cloud can successfully capture the skewness of the probability distribution. The plot also shows the estimated sample correlation coefficient. Since the sample correlation coefficient is calculated without awareness of the semi-definiteness of the joint covariance matrix, it happens to lie outside of the admissible values when a small number of observations are available. The proposed particle filtering method does not suffer from this problem due to constraints on the particle set and works well with few observations. However, the particle filter requires more computational effort than the sample correlation depending on the number of particles.



(a) MSE

(b) ANEES with confidence intervals.

Figure 5.6: Comparison of the sample correlation with the proposed particle filter.

In the following section, the performance of the sample correlation estimation is compared to the proposed particle filtering approach. For this evaluation, an experiment with $N = 100$ observations and MCR = 200 is conducted. Furthermore, the particle filter is used with $M = 250$ (PF20), $M = 1666$ (PF50) and $M = 6796$ (PF100) particles. Figure 5.6a shows the MSE of the experiment. For the particle filters, the mean of the particle set is calculated. The plot shows that the MSE of the particle filters decreases faster than the MSE of the sample correlation. It also shows that more particles result in a better estimation result, although the difference between PF50 and PF100 is insignificant. Furthermore, in Figure 5.6b the credibility of the estimators is evaluated using the ANEES. Since the ANEES requires a covariance matrix, estimates have to be approximated by a Gaussian. In the case of the sample correlation coefficient, the Gaussian approximation of (5.3) is used. For the particle filter, the covariance is calculated from the weighted set of particles. Especially in the beginning with a small number of observations, the uncertainty of the sample correlation is not well approximated, while the particle filters show good results. Again, the number of particles plays a vital role as more particles approximate the uncertainty better and lie within better confidence intervals. The evaluation shows that the proposed particle filter approach shows good performance, both in error and credibility.

5.3 Conclusions to Learning Partial Knowledge about Correlation

This chapter provided a detailed discussion about estimating the correlation coefficients that describe the correlation between estimation errors. Two methods were proposed to estimate the correlation. First, the sample correlation was introduced as an easy way to estimate individual correlation coefficients. The method is well suited to estimate the correlation. However, when only a few observations are available, the probability distribution of sample correlation cannot easily be obtained. Furthermore, since the sample correlation estimates single correlation coefficients, it cannot ensure that the set of coefficients results in a positive semi-definite joint covariance matrix. The second approach introduced is a SIS approach that uses several particles to represent the posterior distribution of the estimated correlation coefficients. This approach allows a flexible parameterization of the correlation matrix and ensures a positive semi-definite joint covariance by using the known constraints of the correlation. Furthermore, the evaluation shows that the particle method is more robust and accurate than the samples correlation coefficient and works well with few observations. It should be noted, that the improved performance might be caused by the particle approach using a model for the estimation and utilizing a prior, while the sample correlation uses only measurements for the estimation. However, the particle approach is computationally expensive, depending on the number of particles. Both results are promising approaches to estimate the correlation of estimation errors. However, this chapter has been limited to examples where the estimates are drawn from a known joint distribution. The results will be applied to a simulation-based approach in the next chapter.

CHAPTER 6

Exploiting Partial Knowledge about Correlation

Contents

6.1 Methods for Information Retrieval 76

6.2 Methods to Exploit Partial Knowledge 78

6.2.1 Bounding Methods Using Partial Knowledge 78

6.2.2 A Gaussian Mixture Approach to Fusion 80

6.3 Parameter Identification for Design of Conservative Bounds 86

6.3.1 Designing Bounds for the Analytic Approach 86

6.3.2 Designing Bounds for the Simulation Approach 89

6.3.3 Discussion 90

6.4 Evaluation 90

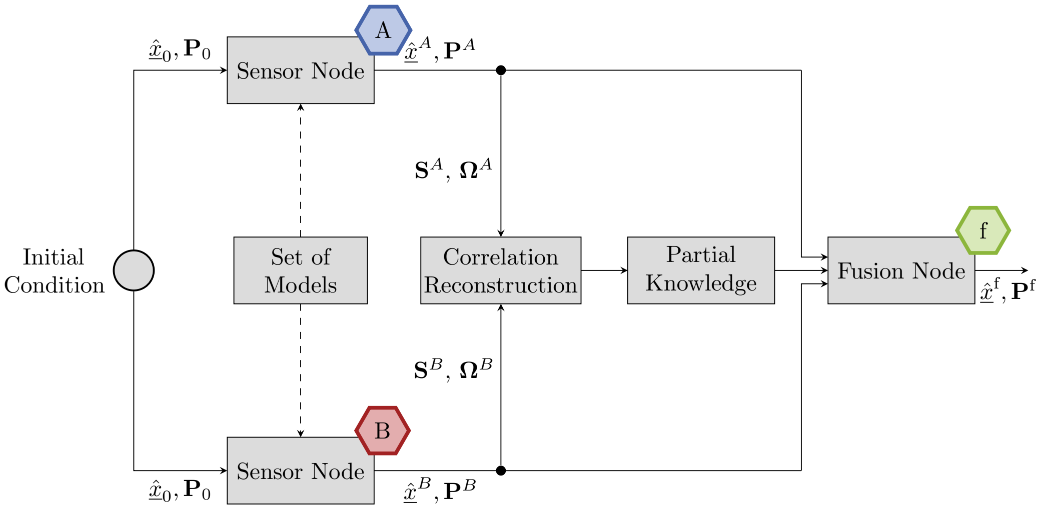
6.4.1 Evaluation of the Simulation Approach 91

6.4.2 Evaluation of the Analytic Approach 93

6.5 Conclusions to Exploiting Partial Knowledge about Correlation 94

As discussed in Chapter 2, there are many reasons why correlations between estimation errors are unknown in many applications. In all cases, unsure or missing knowledge about local processing steps, communication, or the fusion is present. These uncertainties require assumptions that lead to suboptimal fusion results. Based on these previous discussions, this dissertation proposed methods to fully or partially reconstruct cross-covariances that can be used for optimal or suboptimal fusion. Furthermore, Chapter 5 proposed methods to estimate correlations between estimation errors from simulated data. Both approaches are now used as a toolbox to retrieve partial knowledge about correlated estimation errors from sensor networks. Chapter 2 introduced two possible methods to retrieve this partial knowledge, namely an analytic and a simulation approach. During the analytic approach, cross-covariances for a set of possible models are fully or partially reconstructed and normalized to obtain the correlation matrix. During the simulation approach, simulated data of the local estimation steps and the fusion are used to estimate the correlation of the estimation errors. The analytic and the simulation approach result is a set of admissible correlation coefficients that can be exploited during the fusion step.

This chapter proposes solutions for an analytic and a simulation approach. In order to use this knowledge, this chapter begins by investigating suitable bounding methods from the state of the art. Furthermore, a Gaussian Mixture (GM) approach is proposed to utilize retrieved partial knowledge. Afterward, methods to properly parameterize the utilized bounding methods using partial knowledge from the analytic or the simulation approach are proposed. Last, this



A
$\underline{\hat{x}}_0, \mathbf{P}_0$
Sensor Node
$\underline{\hat{x}}^A, \mathbf{P}^A$
$\mathbf{S}^A$, $\mathbf{\Omega}^A$
Initial Condition
Set of Models
Correlation Reconstruction
Partial Knowledge
Fusion Node
f
$\underline{\hat{x}}^{\mathrm{f}}, \mathbf{P}^{\mathrm{f}}$
$\mathbf{S}^B$, $\mathbf{\Omega}^B$
B
Sensor Node
$\underline{\hat{x}}_0, \mathbf{P}_0$
$\underline{\hat{x}}^B, \mathbf{P}^B$

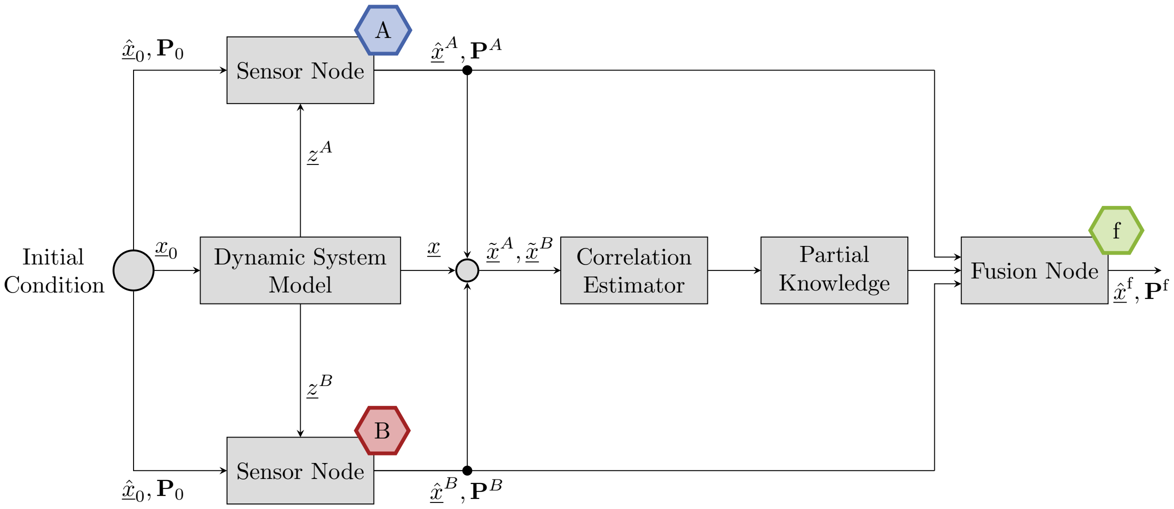
Figure 6.1: Workflow for learning sets of correlation coefficients for two sensor nodes A and B using the analytic approach.

chapter ends with two evaluation examples summarizing the proposed approaches and highlighting advantages and pitfalls.

6.1 Methods for Information Retrieval

The aim of this chapter is to obtain sets of correlation coefficients that characterize the correlation of estimation errors from distributed sensor nodes. Therefore, the following section proposes suitable experiments that are designed to obtain these sets of correlation coefficients. First, the section considers the analytic approach, where the full or partial reconstruction methods from Chapters 3 and 4 are used. Afterward, a suitable setup for the simulation approach is proposed that enables the estimation methods proposed in Chapter 5.

As discussed in Chapter 2, the correlation between estimation errors depends on many factors. For example, system parameters such as system and measurement models, noise covariances for the process, and measurements can change over time and introduce uncertainty about the correlation. Furthermore, the number of sensor nodes, the network topology, and even the method used for the fusion change the correlation. When many parameters can vary, the sets of correlation coefficients could be massive and therefore result in a bound that is not tight and offers only a minimal improvement over conservative fusion methods. Therefore, a suitable set of input parameters to the system is required for which the resulting correlation of estimation errors is learned. Here, the Markov assumption about the system can be utilized. Hence, the output of the local estimators and the correlation between estimation errors only depend on the local estimators' input. In linear systems that are not dependent on the measurements, the input determining the correlation between estimation errors is the initialization of the system with the initial covariance matrix $\mathbf{P}_0$. Therefore, the set of correlation coefficients is learned with a specific initial covariance matrix. This knowledge can then be applied whenever this initial covariance is used. Since the initial covariance is usually not the same during every initialization step, this approach requires learning



Initial Condition
$\underline{x}_0$
$\hat{\underline{x}}_0, \mathbf{P}_0$
Sensor Node
A
$\hat{\underline{x}}^A, \mathbf{P}^A$
$\underline{z}^A$
Dynamic System Model
$\underline{x}$
$\tilde{\underline{x}}^A, \tilde{\underline{x}}^B$
Correlation Estimator
Partial Knowledge
Fusion Node
f
$\hat{\underline{x}}^{\mathrm{f}}, \mathbf{P}^{\mathrm{f}}$
$\underline{z}^B$
B
Sensor Node
$\hat{\underline{x}}_0, \mathbf{P}_0$
$\hat{\underline{x}}^B, \mathbf{P}^B$

Figure 6.2: Workflow for learning sets of correlation coefficients for two sensor nodes A and B using the simulation approach.

sets of correlation coefficients for different initial covariances. Furthermore, since this number of covariances could be enormous, it is necessary to use a suitable grid of initial covariances and interpolate between them using an appropriate distance measure, e.g., [47].

Figure 6.1 shows a possible solution for the analytic approach. When only linear system models are considered, then the simulation of the dynamic system is not necessary because the parameterization of the local estimators does not depend on measurements. Therefore, the correlation between estimation errors only depends on the local models used by the estimators and the initial covariance matrix. This setup is used to fully or partially reconstruct the cross-covariances between the local estimators. For this reconstruction, either a centralized approach or one of the proposed distributed reconstruction methods based on deterministic samples or the square-root decomposition (see Chapters 3 and 4) can be used. This has to be executed for the complete set of possible system models. If this set is too big, a grid of possible models can be used for the reconstruction. If the cross-covariances are fully reconstructed, then the result is a set of discrete correlation coefficients. However, when the cross-covariance is only partially reconstructed until a user-defined time horizon $\mathcal{T}$, the result is a cross-covariance $\mathbf{P}_{\mathcal{T}}^{A,B}$ and a residual term $\mathbf{\Omega}$. Therefore, the normalization yields a set of correlation coefficients with an additional uncertainty. The analytic approach can be used as an offline approach to determine a set of admissible correlation coefficients using a known set of system models. It can also be used while fully or partially reconstructing cross-covariances in a sensor network application to determine the set of correlation coefficients online. The resulting learned correlation can then be used for the fusion instead of further pursuing the reconstruction approach.

While the complete simulation of the local estimators and the fusion step is not required for the analytic approach, it is necessary for the simulation approach. An example for such a setup is shown in Figure 6.2. The Markov assumption of the system is also applied to the simulation approach. Therefore, the local estimators are initialized with a specific initial covariance for which the correlation of estimation errors is learned. Then the full system including the system model, the measurements, the local estimators and the fusion is simulated to obtain local estimates. These can be used to obtain the estimation errors and estimate their correlation. Suitable methods, namely the sample correlation and a particle filtering approach, have been previously introduced in Chapter 5 and are now utilized for the fusion. While the simulation approach can obtain sets of correlated estimation errors online, the setup requires the knowledge of the estimation errors that are usually only available in an offline calibration or simulation routine.

Both methods generate sets of correlation coefficients that need to be accounted for during the fusion step. The next section introduces two different strategies to exploit partial knowledge in the form of sets of correlation coefficients. The first strategy is the use of bounds that obtain a conservative approximation of the uncertainty. Furthermore, suitable methods from the current state of the art are reviewed. The second strategy is to use the sets of correlation coefficients directly using a GM approach.

6.2 Methods to Exploit Partial Knowledge

The following section proposes two approaches to exploit partial knowledge in distributed estimation using known sets of correlation coefficients. The first part of this section is concerned with bounding methods that exploit specific properties of the partial knowledge learned previously. These methods provide a conservative approximation of the uncertainty and are guaranteed to provide credible results. In the second part of this section, a novel GM approach is introduced. This approach can directly use the learned sets of correlation coefficients without bounding, making the fusion result less conservative.

6.2.1 Bounding Methods Using Partial Knowledge

In order to exploit the partial knowledge from the analytic or the simulation approach, the usage of upper bounds for the fusion is a reasonable and well-established choice. The following section is an extended version of the discussion in [150]. Chapter 4 already gave a brief introduction to bounding methods for which CI is the most prominent one. For CI it is assumed that every correlation coefficient in the correlation matrix $\mathbf{\Lambda}$ may lie within their natural bounds $-1 \leq \varrho \leq 1$. However, correlation coefficients are usually smaller than these natural bounds due to the uncorrelated measurement information that the local sensor nodes provide. Therefore, assuming that estimates are fully correlated leads to overly pessimistic estimates that overestimate the uncertainty.

The authors in [7] propose a new general parameterization which was derived from [6, 108, 133]

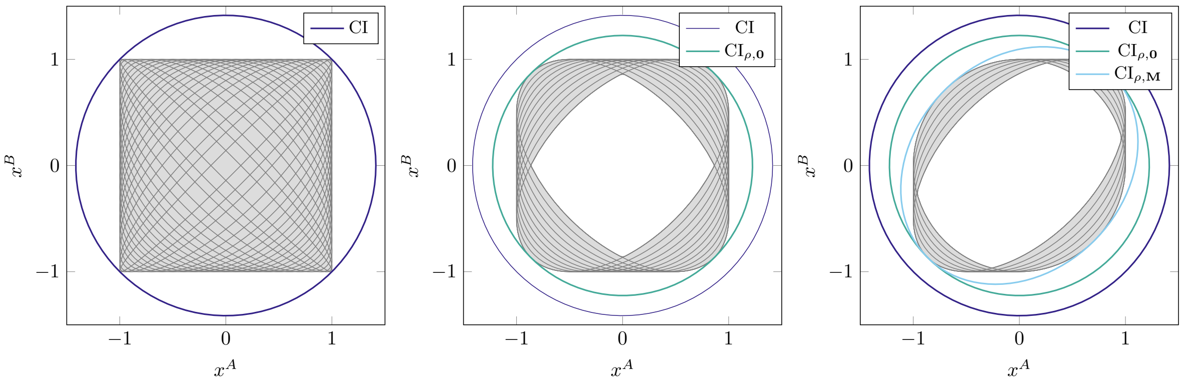
$$\mathbf{J}_{\mathbf{\Lambda}} = \begin{bmatrix} \mathbf{P}^i & \mathbf{D} + \mathbf{\Psi}^i \mathbf{\Lambda} (\mathbf{\Psi}^j)^{\mathrm{T}} \\ \mathbf{D}^{\mathrm{T}} + \mathbf{\Psi}^j \mathbf{\Lambda}^{\mathrm{T}} (\mathbf{\Psi}^i)^{\mathrm{T}} & \mathbf{P}^j \end{bmatrix} ,$$

where the matrix $\mathbf{\Lambda}$ fulfills the condition

$$\forall \mathbf{\Lambda}, \mathbf{I} \geq \mathbf{\Lambda}\mathbf{\Lambda}^{\mathrm{T}} : \mathbf{P}^i \geq (\mathbf{D} + \mathbf{\Psi}^i \mathbf{\Lambda} (\mathbf{\Psi}^j)^{\mathrm{T}}) (\mathbf{P}^j)^{-1} (\mathbf{D} + \mathbf{\Psi}^i \mathbf{\Lambda} (\mathbf{\Psi}^j)^{\mathrm{T}})^{\mathrm{T}} .$$

The centering matrix $\mathbf{D}$ is the middle of admissible values on the cross-covariance $\mathbf{P}^{i,j}$ and the product $\mathbf{\Psi}^i \mathbf{\Psi}^j$ is the radius. Therefore the cross-covariance matrix $\mathbf{P}^{i,j}$ is

$$\mathbf{P}^{i,j} \in \{\mathbf{D} + \mathbf{\Psi}^i \mathbf{\Lambda} (\mathbf{\Psi}^j)^{\mathrm{T}} \,|\, \mathbf{I} \geq \mathbf{\Lambda}\mathbf{\Lambda}^{\mathrm{T}}\} .$$



(a) $-1 \leq \varrho \leq 1$ **(b)** $-0.5 \leq \varrho \leq 0.5$ **(c)** $0 \leq \varrho \leq 0.5$

Figure 6.3: CI under constraints, possible joint covariance matrices (gray).

Furthermore, the authors in [7] propose a family of upper bounds

$$\tilde{\mathbf{J}}_{\mathbf{\Lambda}} = \begin{bmatrix} \mathbf{P}^i + \mu \mathbf{\Psi}^i (\mathbf{\Psi}^i)^{\mathrm{T}} & \mathbf{D} \\ \mathbf{D}^{\mathrm{T}} & \mathbf{P}^j + \frac{1}{\mu} \mathbf{\Psi}^j (\mathbf{\Psi}^j)^{\mathrm{T}} \end{bmatrix} , \tag{6.1}$$

where $0 \leq \mu \leq \infty$ is a positive scalar value that is chosen so to minimize the fused covariance matrix.

A special case of this bound was previously proposed by [108] and simultaneously by [59] in a similar form. [108] proposes a joint covariance matrix

$$\mathbf{J} = \begin{bmatrix} \mathbf{P}^i & \mathbf{P}^{i,j} \\ \mathbf{P}^{j,i} & \mathbf{P}^j \end{bmatrix} \leq \begin{bmatrix} (1+\kappa\rho)\mathbf{P}^i & \mathbf{D} \\ \mathbf{D} & (1+\frac{\rho}{\kappa})\mathbf{P}^j \end{bmatrix} , \tag{6.2}$$

where $\kappa > 0$ is an inflation factor that is optimized so to minimize the trace of the fused covariance matrix. The scalar value ρ is the biggest singular value of the correlation matrix and fulfills

$$\Big[\mathbf{P}^{i,j} - \mathbf{D}\Big]^{\mathrm{T}} (\mathbf{P}^i)^{-1} \Big[\mathbf{P}^{i,j} - \mathbf{D}\Big] \leq \rho^2 \mathbf{P}^j \, .$$

The matrix $\mathbf{D}$ is a centering matrix that can be chosen so that ρ is minimal. In the following discussion of this chapter, this is referred to as CI with scalar parameter and asymmetric constraint. It is also possible to choose the centering matrix as $\mathbf{D} = \mathbf{0}$, referred to as a symmetric constraint. In comparison to the more general bound, $\mathbf{\Psi}^i$ and $\mathbf{\Psi}^j$ are fractions of the covariances $\mathbf{P}^i$ and $\mathbf{P}^j$, e.g., $\alpha\mathbf{P}^i = \mathbf{\Psi}^i(\mathbf{\Psi}^i)^{\mathrm{T}}$ and $\alpha\mathbf{P}^j = \mathbf{\Psi}^j(\mathbf{\Psi}^j)^{\mathrm{T}}$ with any value of $0 \leq \alpha \leq 1$ [7, 66]. The difference of CI and CI with symmetric and asymmetric constraints is illustrated in Example 4. The authors in [134, 66] also consider such a scalar parameter to exploit partial knowledge in distributed estimation.

Example 4: CI bound with scalar parameter and symmetric or asymmetric constraints

A normalized joint covariance matrix of two state estimates $\hat{x}^A$ and $\hat{x}^B$ is assumed to be

$$\mathbf{J} = \begin{bmatrix} \mathbf{S}^A & 0 \\ 0 & \mathbf{S}^B \end{bmatrix} \begin{bmatrix} 1 & \mathbf{\Lambda} \\ \mathbf{\Lambda}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{S}^A & 0 \\ 0 & \mathbf{S}^B \end{bmatrix} ,$$

where $\mathbf{S}^A = \mathbf{S}^B = 1$. Thus, the correlation matrix $\mathbf{\Lambda}$, which needs to fulfill $\mathbf{\Lambda}\mathbf{\Lambda}^{\mathrm{T}} \leq 1$ only includes a single scalar value ϱ. In Figure 6.3a it is assumed that the correlation coefficient lies in its natural constraint $-1 \leq \varrho \leq 1$. All possible correlation coefficients result in a rectangular shape of the joint covariance that is fully bounded by CI. In Figure 6.3b, the correlation coefficient is reduced to $-0.5 \leq \varrho \leq 0.5$. The resulting area of the joint covariance matrix is not rectangular anymore and does not include extreme correlations. Therefore, CI is not tight around the joint covariances, while the symmetric CI bound using $\rho = 0.5$ and $\mathbf{D} = 0$ is a much tighter bound. Finally, the correlation is reduced to $0 \leq \varrho \leq 0.5$ in Figure 6.3c. Both CI and symmetric CI are not tight. However, the asymmetric CI is tight when $\rho = 0.25$ and $\mathbf{D} = 0.25$ are chosen.

The general bound (6.1) and the special case using a scalar value (6.2) are promising candidates to tightly bound sets of correlation coefficients. They require the identification of centering matrices and either scaling matrices or scalar values for the bounds that can be obtained from the analytic or the simulation approach. However, these bounding methods, although tight, still result in conservative approximations of the uncertainty. Therefore, the next section proposes a GM approach to directly use sets of correlation coefficients.

6.2.2 A Gaussian Mixture Approach to Fusion

The following section proposes a fusion rule that uses sets of correlations coefficients that represent the correlation of estimation errors. These sets can also be seen as uncertain correlation coefficients, which can be seen as a probability distribution over the joint probability distribution. Bayesian estimation with uncertain correlation coefficients of joint probability density functions is also considered by the authors in [75]. There, Type 2 densities are defined as parameterized density functions over the state space, whose parameters are described by a density function. The authors in [112] used a set theoretic approach to deal with uncertain correlation coefficients in distributed estimation. Here, it was assumed that no knowledge about the correlation is available, which results in a uniform distribution for the correlation between its natural bounds. The fusion result is a GM, containing all possible solutions of the fusion given the uniformly distributed correlation coefficients. The mean and the covariance of the fused estimate are obtained by moment matching. Furthermore, the results are used to improve the bounds of CI. This approach seems promising. However, the lack of information that motivates a uniform distribution is not a problem anymore, because the analytic or the simulation approach provide additional information that can be exploited. These sets of correlation coefficients are considered in the following to propose a GM approach that is less conservative than the bounds proposed in the previous section.

A Problem Formulation

Two estimates $\hat{\underline{x}}^i$ and $\hat{\underline{x}}^j$ are supposed to be fused into an estimate $\hat{\underline{x}}^{\mathrm{f}}$ with covariance $\mathbf{P}^{\mathrm{f}}$. The correlations between the estimation errors of $\hat{\underline{x}}^i$ and $\hat{\underline{x}}^j$ are partially known and the joint covariance matrix is denoted by

$$\mathbf{J} = \begin{bmatrix} \mathbf{P}^i & \mathbf{P}^{i,j} \\ \mathbf{P}^{j,i} & \mathbf{P}^j \end{bmatrix} .$$

The cross-covariance matrix can be calculated by

$$\mathbf{P}^{i,j} = \mathbf{S}^i \mathbf{\Lambda} (\mathbf{S}^j)^{\mathrm{T}} ,$$

where $\mathbf{S}^i$ and $\mathbf{S}^j$ denote the Cholesky decompositions of the local covariances $\mathbf{P}^i$ and $\mathbf{P}^j$. Furthermore, the matrix $\mathbf{\Lambda}$ describes the correlation of estimation errors of $\hat{\underline{x}}^i$ and $\hat{\underline{x}}^j$ and it contains several correlation coefficients that can be listed in a parameter vector $\mathrm{vec}(\mathbf{\Lambda}) = \underline{\lambda} = [\varrho_1, \dots, \varrho_{n_\varrho}]^{\mathrm{T}}$, where n_ϱ is the number of correlation coefficients included in the correlation matrix. If the correlation matrix $\mathbf{\Lambda}$ is known, then the fusion can be executed according to Bar-Shalom–Campo (BSC) formulas (see (2.10)).

However, the correlation is only partially known. It is assumed that the parameter vector $\underline{\lambda}$ contains random variables from a stochastic process and that those correlation coefficients have an arbitrary pdf. The aim is to calculate the pdf of the fused estimate $f(x^{\mathrm{f}})$ depending on the correlation coefficient

$$f(x^{\mathrm{f}}) = \int f(x^{\mathrm{f}}, \underline{\lambda}) \, \mathrm{d}\underline{\lambda} = \int f(x^{\mathrm{f}} \,|\, \underline{\lambda}) f(\underline{\lambda}) \, \mathrm{d}\underline{\lambda} \tag{6.3}$$

with the density of the fusion result depending on the correlation coefficient $f(x^{\mathrm{f}} \,|\, \underline{\lambda}) = f(\hat{\underline{x}}^{\mathrm{f}}, \mathbf{P}^{\mathrm{f}} \,|\, \underline{\lambda})$. To solve this equation, (6.3) needs to be marginalized. Since this is not possible in closed form, the result can be approximated with one Gaussian or a mixture.

The authors in [112] assumed to be agnostic about the correlation coefficient. Since there is no knowledge about the correlation given, the correlation coefficients lie in the interval between $[-1, 1]$. However, partial knowledge is now available and can be used to expand this GM approach. It is assumed that a set of correlation coefficients $\{\underline{\lambda}\}_{m=1}^{M}$ is sampled from the distribution of the correlation. For example, this distribution could be directly obtained from the particle filter proposed in Chapter 5. For the fusion result, a GM is obtained

$$\begin{aligned} f(x^{\mathrm{f}}) &= \int f(x^{\mathrm{f}}, \underline{\lambda}) \, \mathrm{d}\underline{\lambda} = \int f(x^{\mathrm{f}} \,|\, \underline{\lambda}) f(\underline{\lambda}) \, \mathrm{d}\underline{\lambda} \\ &= \sum_{m=1}^{M} \omega_m \cdot f(x^{\mathrm{f}} \,|\, \underline{\lambda}_m) , \end{aligned}$$

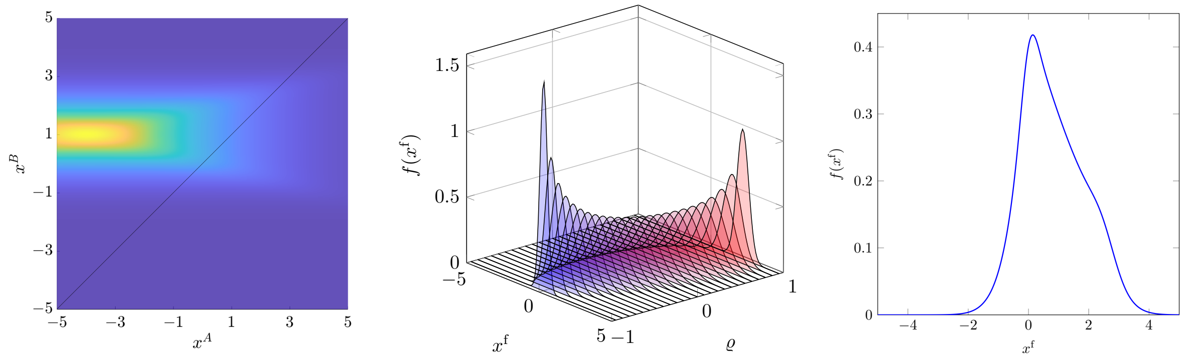
with $\sum_{m=1}^{M} \omega_m = 1$ and M mixture components. The weights of the mixture components are calculated using

$$\begin{aligned} \mathbf{S} &= \mathbf{P}^i + \mathbf{P}^j - \mathbf{P}^{i,j} - \mathbf{P}^{j,i} \\ \omega_m &= \frac{1}{\sqrt{2\mathbf{S}}} \exp\Big(- \frac{1}{2} \frac{(\hat{\underline{x}}^i - \hat{\underline{x}}^j)^2}{\mathbf{S}} \Big) . \end{aligned}$$

Therefore, for every $\underline{\lambda}_m$ a fusion result using the BSC formulas (see Figure 6.4) is generated. An illustration for this approach is given in the following example.

Example 5: Fusion of Two One-dimensional State Estimates Using Gaussian Mixtures
Two one-dimensional state estimates from node A and B are given by $\hat{\underline{x}}^A = 1$, $\hat{\underline{x}}^B = -4$, $\mathbf{P}^A = 1$, $\mathbf{P}^B = 4$. Furthermore, it is assumed that the set of possible correlation coefficients contains $M = 25$ evenly spaced correlation coefficients from $\varrho_{\min} = -0.95$ to $\varrho_{\max} = 0.95$.

For every correlation coefficient, a fusion result is calculated as can be seen in Figure 6.4b. Moreover, it is possible to plot the superposition of the realizations of the joint covariance matrix with every



(a) Joint density of mixture.

(b) Mixture components using BSC formulas for every correlation coefficient.

(c) Resulting GM density.

Figure 6.4: Visualization of the joint density, the mixture components and the resulting fused density using the GM approach.

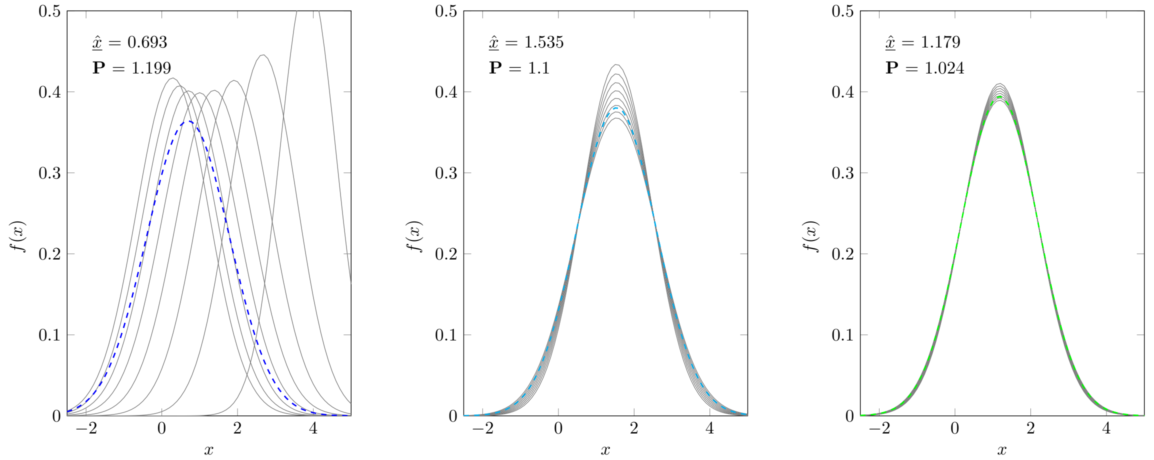
correlation coefficient, which can be seen in Figure 6.4a. This joint probability distribution is not Gaussian distributed anymore [59]. The fusion is equivalent to a cut through this joint probability distribution on the diagonal [97]. It can be seen that the resulting GM density (see Figure 6.4c) of realizations of the fusion with the correlation coefficients is also not Gaussian. Furthermore, it is tilted towards the negative correlation coefficient, introducing a bias towards negative correlations.

B Discussion

Several aspects of this GM approach have to be taken into consideration. The first problem that can arise is that there is no or very uncertain knowledge about the correlation between estimation errors. Whenever there is only little knowledge to exploit, then the fusion result can become biased due to negative correlation coefficients that have a strong influence on the GM. At the same time, the positive correlation has a minor impact. Using partial knowledge about the correlation reduces the bias, as extreme correlation coefficients are usually not included, and the uncertainty of the correlation is also reduced.

The second problem is that there is a true correlation between estimation errors for that the fusion result can be calculated optimally. While the mixture includes the correct fusion result, all other fusion results are incorrect to some degree. Thus, the mixture can be seen as several hypotheses about the correlation matrix. When the mixture is used to reinitialize the local estimators, these hypotheses can be tested whether they are consistent with the local observations. Therefore, the hypothesis tree can be pruned until the true correlation matrix is found.

However, processing the complete mixture in the local estimators can be cumbersome or infeasible because of limited bandwidth or local computational power. Thus, the mixture could be approximated by a single Gaussian. This mixture reduction can be problematic because it does not catch the true density or its moments. Because of the shape of the density (see Figure 6.4), the resulting Gaussian can underestimate the uncertainty and might therefore lead to noncredible local estimators. A possible approach to reduce the risk of underestimating the uncertainty is to



(a) BSC for every mixture component individually.

(b) Average the fusion gains and solve the BSC formulas.

(c) Average joint covariance matrix, then solve the BSC formulas.

Figure 6.5: Comparison of different weighting techniques for the GM approach, mixture components (gray) and approximation using single Gaussian (colored).

approximate the GM not by one but a few Gaussian using a clustering algorithm, e.g., Expectation Maximization [38, 51]. Another way could be to reduce the GM as late as possible, e.g., before communicating with the fusion center again. Then, the hypothesis tree is already pruned, and only a couple of mixture components remain, making the approximation much more credible.

The last problem considered is the calculation of the fusion gain. Previously it was only discussed that the BSC formulas are applied. However, there are several solutions to finding the fusion gain:

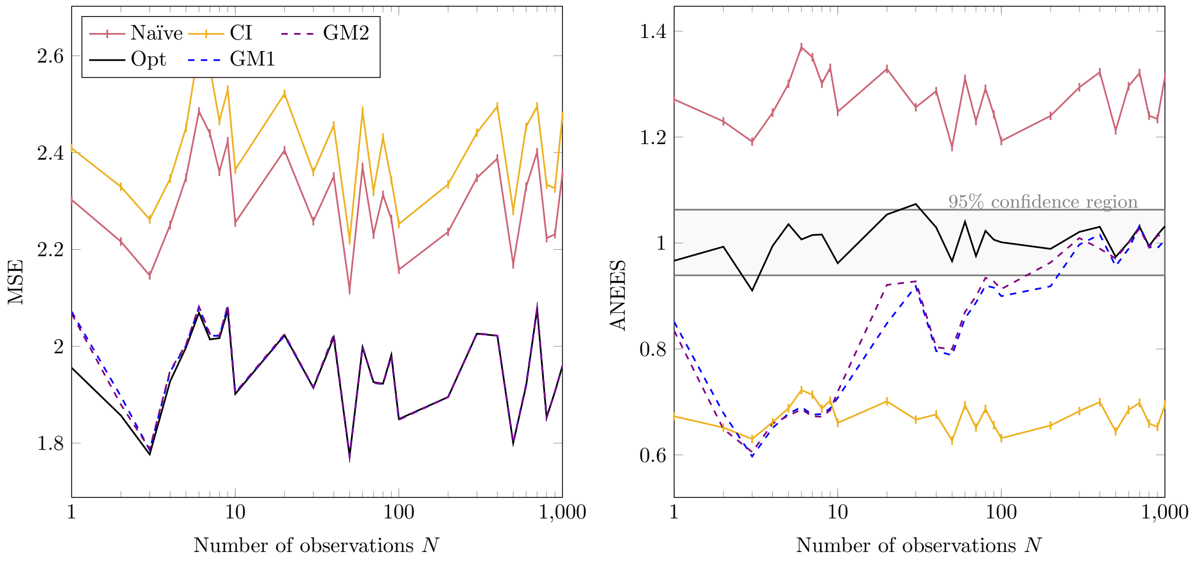
1. solve the BSC formulas for every mixture component individually,
2. calculate the fusion weights for every component individually, then average the fusion gains and solve BSC formulas with the averaged fusion gain, or
3. calculate the average joint covariance matrix, then solve BSC formulas with the fusion gain for the average joint covariance matrix.

These approaches for the fusion lead to very different fusion results as can be seen in the following example.

Example 6: GM Approach using Different Methods to Calculate the Fusion Gain

Again, two one-dimensional state estimates with $\mathbf{P}^A = 1$, $\mathbf{P}^B = 4$, $\hat{\underline{x}}^A = 1$, $\hat{\underline{x}}^B = -4$ *are to be fused. It is assumed that the set of possible correlation coefficients contains* $M = 8$ *evenly spaced correlation coefficients from* $\varrho_{\min} = 0.2$ *to* $\varrho_{\max} = 0.9$*.*

Figure 6.5 shows the mixture components and the fusion result approximated by a single Gaussian using moment matching. Solving the BSC formulas for every component leads to bigger variance, while the smallest variance results from the weighting of the average joint covariance. Since a big concern is that the uncertainty might be underestimated, the element-wise solving of the BSC-formulas seems to be more reasonable. In the following section, an experiment to evaluate



(a) MSE for 1000 MCRs.

(b) ANEES for 1000 MCRs.

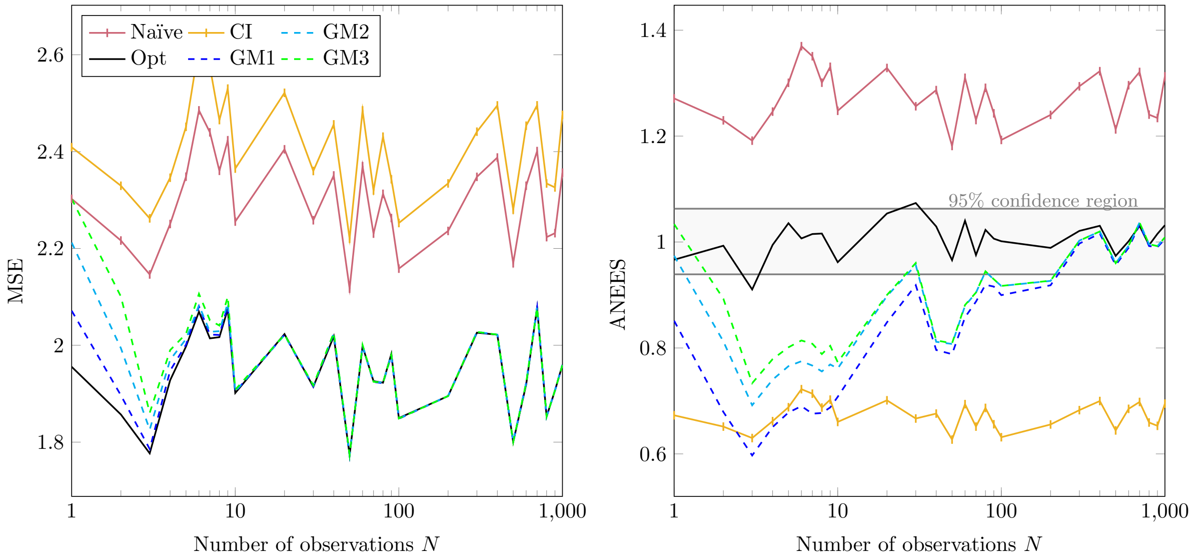
Figure 6.6: Evaluation of the GM approach with different number of particles.

the performance of the proposed GM approach with different parameterization is conducted and the results are compared to other fusion methods.

C Evaluation

For the evaluation of the proposed GM approach, the simulation from Section 5.2.2.D is used, where the proposed particle filter approach was utilized to estimate the correlation matrix with the true values $\mathbf{\Lambda} = [0.68, 0.68]^\mathrm{T}$. The particle filter is used in two configurations, with $M = 284$ (GM1) and $M = 1224$ (GM2) particles. The estimation of the correlation is run for $N = 1000$ observations. In the first part of this evaluation, the GM approach uses the particle filters directly to solve the BSC formulas for every sample from the estimated correlation matrix. Therefore, the GM includes 284, or 1224, mixture components, respectively. In order to compare the mixture with the other fusion methods, the mixture is approximated by a single Gaussian using moment matching. The fusion results are compared with the naïve fusion, CI, and the optimal fusion (BSC) using the true correlation matrix. The fusion step is only evaluated at certain time steps closer together at the beginning and sparser later on. Random samples from the true joint covariance are drawn and fused with the competing fusion methods. Furthermore, 1000 MCRs are executed for every fusion step.

Figure 6.6a shows the MSE of the fused estimates. The x-axis is shown on a logarithmic scale to better visualize the uneven spacing of the fusion steps. The later the fusion step occurs, the more observations have been used to estimate the particle set and the better the estimate of the correlation. While naïve fusion and CI show slightly higher MSE than the optimal fusion, the GM approach is indistinguishable to the optimal fusion result after about $N = 10$. Figure 6.6b evaluates the credibility of the fusion result using the ANEES. CI is rather conservative and outside of the 95% confidence interval, and naïve fusion is also mostly outside of the 95% confidence interval.



(a) MSE for 1000 MCRs.

(b) ANEES for 1000 MCRs.

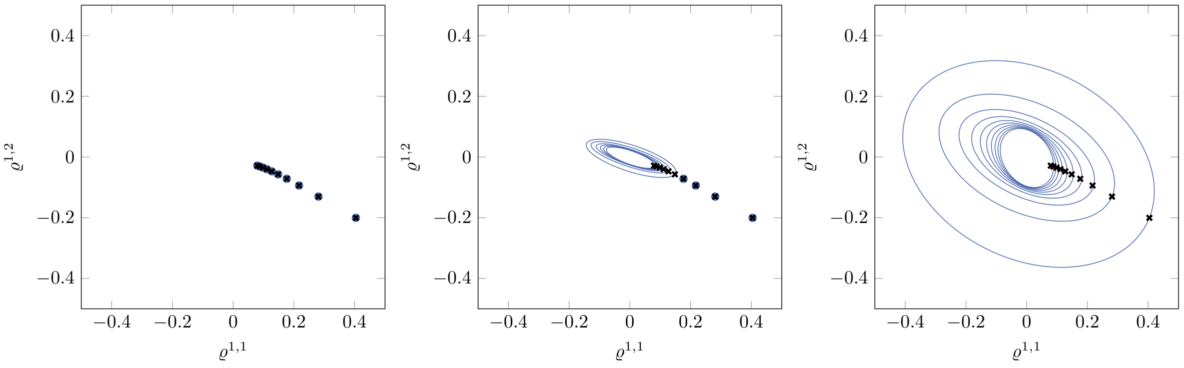
Figure 6.7: Evaluation of the GM approach using different calculation methods of the fusion gain.

Nevertheless, the GM approach is very close to CI in the beginning and then approaches the result of the optimal fusion that it is almost identical to after a while. The number of particles seems to have no significant influence on the quality of the fusion result. Overall, the GM approach shows excellent fusion results and seems to be a reasonable choice for the fusion. It is convenient since it can directly use the estimated point cloud from the correlation estimator for the fusion. Next, the different methods for calculating the fusion gain are evaluated. This time, all GM methods use $M = 284$ particles. However, the approach was used in the following configurations:

- solve BSC formulas for every mixture component individually (GM1),
- calculate the fusion weights for every component individually, then average the fusion gains and solve BSC formulas with the averaged fusion gain (GM2) and
- calculate the average joint covariance matrix, then solve BSC formulas with the fusion gain for the average joint covariance matrix (GM3).

The MSE in Figure 6.6a is higher for the methods with averaging of weights of the joint covariance matrix then for the element-wise solving of the BSC formulas. However, the methods are also significantly less conservative (see Figure 6.6b) with small number of observations.

To summarize, the proposed GM approach shows good performance. The approach tends to be more conservative when the uncertainty of the correlation estimate is high. However, as soon as the uncertainty decreases significantly, the performance is close to the optimal fusion. The weighting scheme seems to have a high impact on the fusion result. The element-wise solution of the BSC formulas seems to be more conservative but also results in smaller errors. Future research is necessary to investigate whether this approach is practical in real-life applications. Furthermore, the GM approach is only applied to the fusion step. Therefore, it is still unclear how the local estimators can further process the result and how this influences the correlation estimate.



(a) Full reconstruction. **(b)** Partial reconstr. ($\mathcal{T} = 5$). **(c)** Partial reconstr. ($\mathcal{T} = 1$).

Figure 6.8: Visualization of full and partial analytic calculation of the cross-correlation, actual cross-correlation (black cross), reconstructed set of cross-correlations (blue).

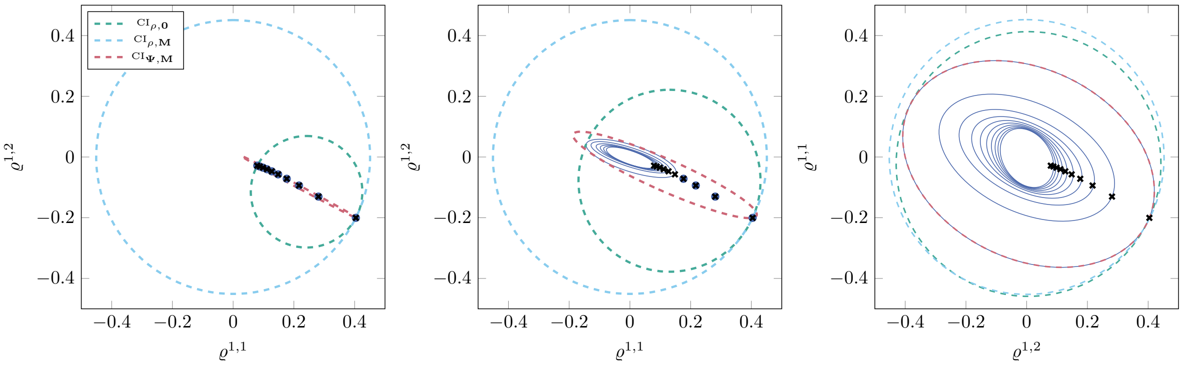
6.3 Parameter Identification for Design of Conservative Bounds

Section 6.1 discussed how sets of correlation coefficients can be generated by simulation or reconstruction. Section 6.2 proposed different approaches to utilize these sets of correlation coefficients for the fusion, where the first approach is the use of bounding techniques. Two possible bounding approaches have been identified that are promising candidates for exploiting sets of correlation coefficients. The first approach, which uses a family of bounds as proposed in [7] requires a centering matrix and, additionally, two scaling matrices $\mathbf{\Psi}^i$ and $\mathbf{\Psi}^j$. The second bounding method proposed by [108] only needs a scalar parameter ρ and a centering matrix $\mathbf{D}$, and is a special case of the more general bound. However, the missing link is the parameterization of these bounding methods since they require specific matrices or scalar parameters to be extracted from the set of correlation coefficients. The following section proposes strategies to extract these parameters from the set of correlation coefficients to use them during the fusion. Moreover, the analytic and the simulation approach require different approaches since their sets correspond to different levels of information. For simplicity, the discussion is limited to the parameter identification for sets of correlation coefficients that contain only two correlation coefficients and therefore lie in $\mathbb{R}^2$ since this is easier to visualize.

6.3.1 Designing Bounds for the Analytic Approach

The analytic approach obtains sets of cross-covariances. To transform these to correlation coefficients, they have to be normalized. A suitable decomposition of the system is given in (5.2), resulting in a correlation matrix $\mathbf{\Lambda}$. If the reconstruction is only partially executed, then an additional residual term that represents the uncertainty of the cross-covariance matrix is generated. When this residual is used, a joint covariance of the form

$$\mathbf{J} = \begin{bmatrix} \mathbf{P}^i & \mathbf{P}^{i,j}_{\mathcal{T}} + \mathbf{\Omega}^i \mathbf{\Lambda} (\mathbf{\Omega}^j)^{\mathrm{T}} \\ \mathbf{P}^{j,i}_{\mathcal{T}} + \mathbf{\Omega}^j \mathbf{\Lambda}^{\mathrm{T}} (\mathbf{\Omega}^i)^{\mathrm{T}} & \mathbf{P}^j \end{bmatrix} \quad , \quad \mathbf{\Lambda}\mathbf{\Lambda}^{\mathrm{T}} \leq \mathbf{I}$$



(a) Full reconstruction. **(b)** Partial reconstr. ($\mathcal{T}=5$). **(c)** Partial reconstr. ($\mathcal{T}=1$).

Figure 6.9: Fitting of bounding circles and ellipses for the analytic approach with full and partial reconstruction, actual cross-correlation (black cross), reconstructed set of cross-correlations (blue).

is constructed. This is a joint covariance of the same form as in (6.1). Therefore $\mathbf{P}_{\mathcal{T}}^{i,j}$ is equal to the center matrix $\mathbf{D}$ and $\mathbf{\Omega}^i(\mathbf{\Omega}^j)^{\mathrm{T}}$ is the radius of an ellipse. Hence, the data from the analytic approach contains correlation coefficients for fully reconstructed normalized cross-covariances and sets of correlation coefficients for partially reconstructed normalized cross-covariances and their residual term. Hence, the set of correlation coefficients is mix of data, especially when a fixed time horizon allows full reconstruction for some models and partial reconstruction for others. The following example visualizes this problem of full and partial reconstruction and the resulting sets of correlation coefficients for different parameterizations of the time horizon.

Example 7: Cross-Correlation Sets for Full and Partial Reconstruction

Two sensor node A and B are estimating the state of a discrete-time time-invariant linear stochastic system with

$$\mathbf{A}=\begin{bmatrix}1 & \Delta T\\ 0 & 1\end{bmatrix}, \ \mathbf{Q}=\begin{bmatrix}\frac{\Delta T^3}{3} & \frac{\Delta T^2}{2}\\ \frac{\Delta T^2}{2} & \Delta T\end{bmatrix}, \ \Delta T=1\,.$$

Both senor nodes use a linear measurement model with measurement matrices $\mathbf{C}^A=\mathbf{C}^B=\mathbf{I}$ *and measurement covariances* $\mathbf{R}^A=\mathrm{diag}(5^2,1^2)$ *and* $\mathbf{R}^B=\mathrm{diag}(2^2,1^2)$*. Node A sends only its first state* $\underline{\hat{x}}_1^A$ *and the belonging variance* $\mathbf{P}_{11}^A$ *and node B sends the full state estimate* $\underline{\hat{x}}^B$ *and covariance* $\mathbf{P}^B$ *to the fusion center. It is assumed that the fusion could be executed somewhere between the 1st or 10th time step. The reconstructing of the cross-covariance matrix is executed using the square-root decomposition as discussed in Chapter 3. Furthermore, the time horizon is limited to* $\mathcal{T}=5$ *and* $\mathcal{T}=1$ *to create a residual term as discussed in Chapter 4.*

Figure 6.8 shows the reconstruction of the cross-correlation, which is obtained by normalizing the reconstructed set of cross-covariance matrices. The plot on the left shows the full reconstruction of the cross-covariances. The middle and the right plot show the partial reconstruction It can be observed that the smaller the time horizon, the bigger the surrounding ellipse that contains the set of possible correlation coefficients.

The set of correlation coefficients caused by the partial reconstruction can also be replaced by a set of correlation coefficients that form the bounding ellipse. To reduce the amount of data, one could only save the data points of the convex hull, making the parameter identification for the bounding faster. In the analytic approach, a set of possible correlation matrices is generated. Furthermore, it is assumed that every set of correlation matrices is possible and could theoretically occur. Therefore, the full set needs to be bounded. The following paragraphs propose methods to calculate bounding ellipses or circles that enclose the set of possible correlation coefficients.

Designing Bounds with Matrices $\boldsymbol{\Psi}$ and Centering Matrix $\mathbf{D}$: When the correlation matrix only contains two correlation coefficients, then the set of correlation coefficients describes an ellipse in $\mathbb{R}^2$ of the form

$$\mathcal{E}_{\Theta,\underline{\mu}}\{\underline{\lambda} \in \mathbb{R}^2 : (\underline{\lambda} - \underline{\mu})^{\mathrm{T}} \Theta^{-1} (\underline{\lambda} - \underline{\mu})\}\,,$$

where $\underline{\lambda} = \mathrm{vec}(\boldsymbol{\Lambda})$ and where $\underline{\mu} = \mathrm{vec}(\mathbf{M})$ is the center of the ellipse. This center can be calculated by normalizing a centering matrix $\mathbf{M} = (\mathbf{S}^i)^{-1}\mathbf{D}\big((\mathbf{S}^j)^{-1}\big)^{\mathrm{T}}$. Furthermore, the covariance matrix is $\Theta = \boldsymbol{\Phi}^i(\boldsymbol{\Phi}^i)^{\mathrm{T}}\boldsymbol{\Phi}^j(\boldsymbol{\Phi}^j)^{\mathrm{T}}$, where $\boldsymbol{\Phi}$ is the normalized scaling matrix is $\boldsymbol{\Phi} = (\mathbf{S}^i)^{-1}\boldsymbol{\Psi}\big((\mathbf{S}^j)^{-1}\big)^{\mathrm{T}}$. A set of correlation matrices was obtained from the analytic approach

$$\{\underline{\lambda}\}_{m=1}^{M} = [\underline{\lambda}_1, \ldots, \underline{\lambda}_M]\,.$$

The aim is to minimize the area of the ellipse, e.g., by minimizing the trace of the covariance Θ

$$\begin{aligned} \underset{\Theta,\underline{\mu}}{\arg\min} \quad & \det(\Theta) \\ \text{s.t.} \quad & (\underline{\lambda}_m - \underline{\mu})^{\mathrm{T}}\Theta^{-1}(\underline{\lambda}_m - \underline{\mu}) \leq 1 \quad , \quad \forall m = 1, \ldots, M\,. \end{aligned}$$

There are several publications concerned with this problem of finding the optimal ellipse or ellipsoid with the minimum area or volume [78, 126]. Here, the Matlab implementation from [93] and their documentation is used. The center of the ellipse $\underline{\mu}$ is relatively easy to find. However, the matrices $\boldsymbol{\Psi}^i$ and $\boldsymbol{\Psi}^j$ are not unique, as already discussed in [7]. There are a number of possible combinations of $\boldsymbol{\Psi}^i$ and $\boldsymbol{\Psi}^j$ that correspond to the same set of correlation coefficients. There are relatively little constraints to the search. The number of rows has to correspond to the sizes of $\mathbf{P}^i$ and $\mathbf{P}^j$ and $\boldsymbol{\Psi}^i$ and $\boldsymbol{\Psi}^j$ need to have full column-rank. In the case of the ellipse, the parameter identification can be designed as follows. If the dimension of $\boldsymbol{\Psi}^i$ is 1 and the dimension of $\boldsymbol{\Psi}^j$ is 2, then matrix $\boldsymbol{\Psi}^i := \sqrt{1/\beta}$ and $\boldsymbol{\Psi}^j := \sqrt{\beta\Theta}$, where β is an arbitrary scaling factor.

Designing Bound with Scalar Value ρ and Centering Matrix $\mathbf{D}$: The identification of the scalar value ρ is also discussed in [108]. The aim is to minimize the biggest singular value of the set of correlation matrices [134]. Therefore, a normalized centering matrix $\mathbf{M}$ is required that leads to a center of the ellipse $\underline{\mu} = \mathrm{vec}(\mathbf{M})$

$$\mathcal{E}_{\rho,\underline{\mu}}\{\underline{\lambda} \in \mathbb{R}^2 : (\underline{\lambda} - \underline{\mu})^{\mathrm{T}}(\rho^2\mathbf{I})^{-1}(\underline{\lambda} - \underline{\mu})\}$$

that minimizes the biggest singular value ρ. In the case where a correlation matrix with two coefficients is assumed, the ellipse is reduced to a circle in $\mathbb{R}^2$ with center $\underline{\mu}$ and radius ρ, where the scalar value $\rho \leq 1$. The optimization problem can be posed like this

$$\underset{\rho,\underline{\mu}}{\arg\min} \left(\max_{\rho} \left(\mathrm{svd}(\underline{\lambda}_m - \underline{\mu}) \right) \right).$$

The problem can even be simplified by assuming a symmetric constraint of the bound. Then, the centering matrix $\mathbf{D} = \mathbf{0}$ and the aim is to only find the biggest singular value, which can be done without any optimization. Figure 6.9 shows the proposed bounds for the Example 7.

6.3.2 Designing Bounds for the Simulation Approach

For the analytic approach, a set of correlation coefficients is obtained that do not represent a pdf, but rather possible outcomes of the estimation process that all, theoretically, can occur. Therefore, the complete set needs to be bounded. However, during the simulation approach, a set of correlation coefficients is obtained representing a probability distribution of correlation coefficients, e.g., a particle cloud. Then, a confidence interval has to be generated to bound this distribution that includes a certain probability mass. The higher the probability mass included, the more conservative the fusion result.

Designing Bound with Matrices Ψ and Centering Matrix D: Let $\{\underline{\lambda}\}_{m=1}^{M}$ be a set of M particles obtained from the simulation approach with a belonging set of weights $\{\omega\}_{m=1}^{M}$. The mean of the particle set can be calculated according to

$$\bar{\underline{\lambda}} = \sum_{m=1}^{M} \omega_m \underline{\lambda}_m \,.$$

This particle mean $\bar{\underline{\lambda}}$ is assumed to include the parameters of the normalized centering matrix $\mathbf{M}$ with $\bar{\underline{\lambda}} = \mathrm{vec}(\mathbf{M})$. Furthermore, the covariance of the particle set is calculated

$$\mathbf{P} = \sum_{m=1}^{M} \omega_m \big(\underline{\lambda}_m - \bar{\underline{\lambda}}\big)\big(\underline{\lambda}_m - \bar{\underline{\lambda}}\big)^{\mathrm{T}} \,.$$

Then, assuming a Gaussian distribution, the bounding ellipse is calculated using an appropriate confidence interval, e.g., the 95% confidence interval

$$\Theta = 1.96^2\,\mathbf{P} \,.$$

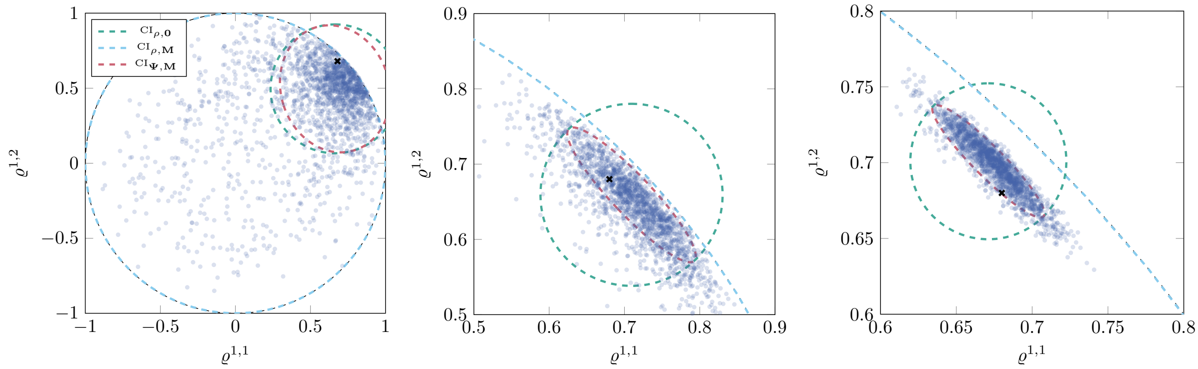
From this ellipse, the matrices $\boldsymbol{\Psi}^i$ and $\boldsymbol{\Psi}^j$ can be calculated as proposed in 6.3.1.

Designing Bound with Scalar Value ρ and Centering Matrix D: For the calculation of the scalar value ρ, the biggest singular value of the covariance $\mathbf{P} = \sum_{m=1}^{M} \omega_m \big(\underline{\lambda}_m - \bar{\underline{\lambda}}\big)\big(\underline{\lambda}_m - \bar{\underline{\lambda}}\big)^{\mathrm{T}}$ has to be computed. Then, assuming a Gaussian distribution, an appropriate confidence interval is chosen, e.g., the 95% confidence interval, to calculate

$$\rho = 1.96\sqrt{\max(\mathrm{svd}(\mathbf{P}))} \,.$$

If the centering matrix is chosen as $\mathbf{D} = \mathbf{0}$, then the covariance is calculated without using the mean of the particle set according to $\mathbf{P} = \sum_{m=1}^{M} \omega_m \big(\underline{\lambda}_m\big)\big(\underline{\lambda}_m\big)^{\mathrm{T}}$. Again, the 95% confidence interval is computed to obtain the biggest singular value. When the uncertainty of the particle set is big, then there are cases where the scalar value ρ can be bigger than 1. However, it makes sense to limit the value to 1, since bigger values are not admissible and lead to overbounding of the uncertainty.

A particle cloud from the simulation approach (see Section 5.2.2.D) is considered for different number of observations. The results of the proposed bounding approach are shown in Figure 6.10. Because the particle cloud is close to the natural constraint of the correlation, the ellipse using the scaling matrices is a smaller bound around the particles than the bounds using scalar values. Furthermore, using the centering matrix unequal to zero is advantageous, resulting in a smaller bound. When the symmetric constraint is chosen, there is no improvement compared to CI. The



(a) $N = 5$.

(b) $N = 25$.

(c) $N = 100$.

Figure 6.10: Fitting of bounding circles and ellipses for the simulation approach.

plots also show that the bounded area becomes smaller the more observations are available. Last, it can be seen that these bounding techniques lead to bounds that can lie outside of the admissible values and this is much more likely with the bounding techniques using scalar values.

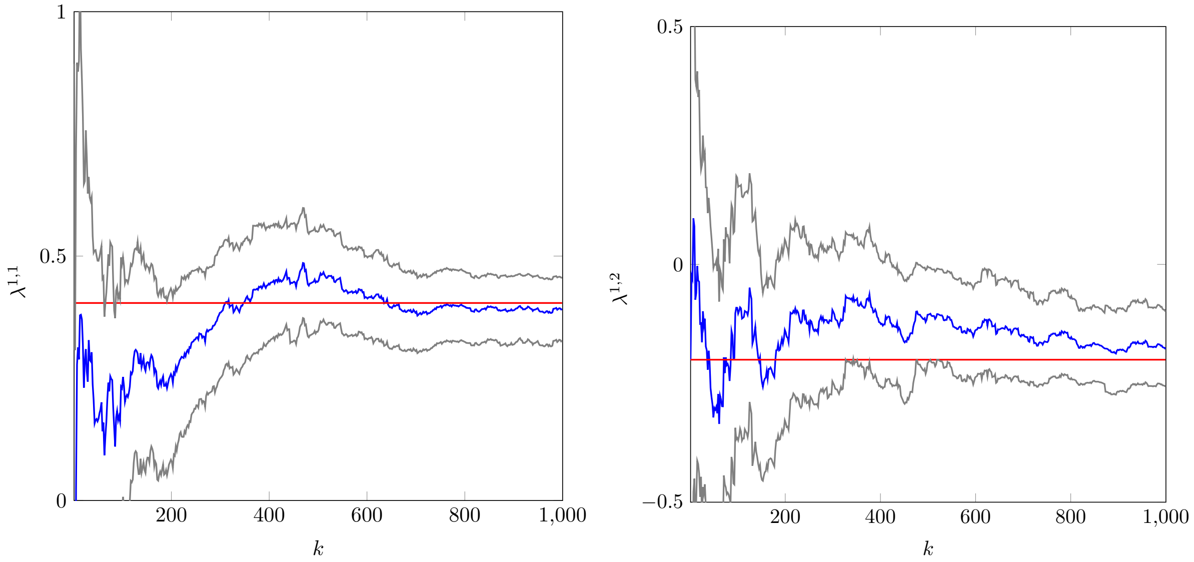
6.3.3 Discussion

The decomposition to obtain the scaling matrices $\boldsymbol{\Psi}$ for the bound proposed in [7] is not unique. In the case that only two correlation coefficients are within the correlation matrix, this is unproblematic since the upper bound calculation uses a scalar value that cancels out the parameterization of the matrices. However, this might not be the case in higher-dimensional problems where $n_\varrho > 2$. Furthermore, in higher-dimensional cases, the bound is not an ellipse but a more complicated shape (as discussed in Chapter 5). Therefore, further research is necessary to find appropriate optimization techniques for the scaling matrices and the centering matrix. However, the bound using the scalar value does not suffer from this problem, and the proposed bounding technique can easily be generalized to higher-dimensional problems. Another problem that becomes apparent in Figure 6.10 is that the derived bounds may include correlation coefficients that are not admissible since they are outside of the constraints. While mathematically perfectly fine, these correlation coefficients could lead to overly conservative fusion results. Further research is necessary to investigate the degree of performance loss.

6.4 Evaluation

The following section evaluates the analytic and the simulation approach. Both evaluation examples assume the same system, where two sensor nodes A and B estimate the state of a discrete-time time-invariant linear stochastic system with

$$\mathbf{A} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix}, \quad \mathbf{Q} = 0.1 \begin{bmatrix} \frac{\Delta T^3}{3} & \frac{\Delta T^2}{2} \\ \frac{\Delta T^2}{2} & \Delta T \end{bmatrix}, \quad \Delta T = 0.1\,.$$



(a) Estimated correlation coefficient $\lambda^{1,1}$.

(b) Estimated correlation coefficient $\lambda^{1,2}$.

Figure 6.11: Estimated correlation coefficients $\lambda^{1,1}$ and $\lambda^{1,2}$ (blue) and the estimated 95% confidence interval ϱ (gray) with increasing number of training covariances compared to the analytically calculated correlation coefficients $\varrho^{1,1}$ and $\varrho^{1,2}$ (red) for a single training covariance.

Both senor nodes use a linear measurement model with measurement matrices $\mathbf{C}^A = \mathbf{C}^B = \mathbf{I}$, where the observations are corrupted by additive white Gaussian noise with covariances

$$\mathbf{R}^A = \mathrm{diag}(5^2, 1^2)\,, \ \mathbf{R}^B = \mathrm{diag}(0.5^2, 0.5^2)\,.$$

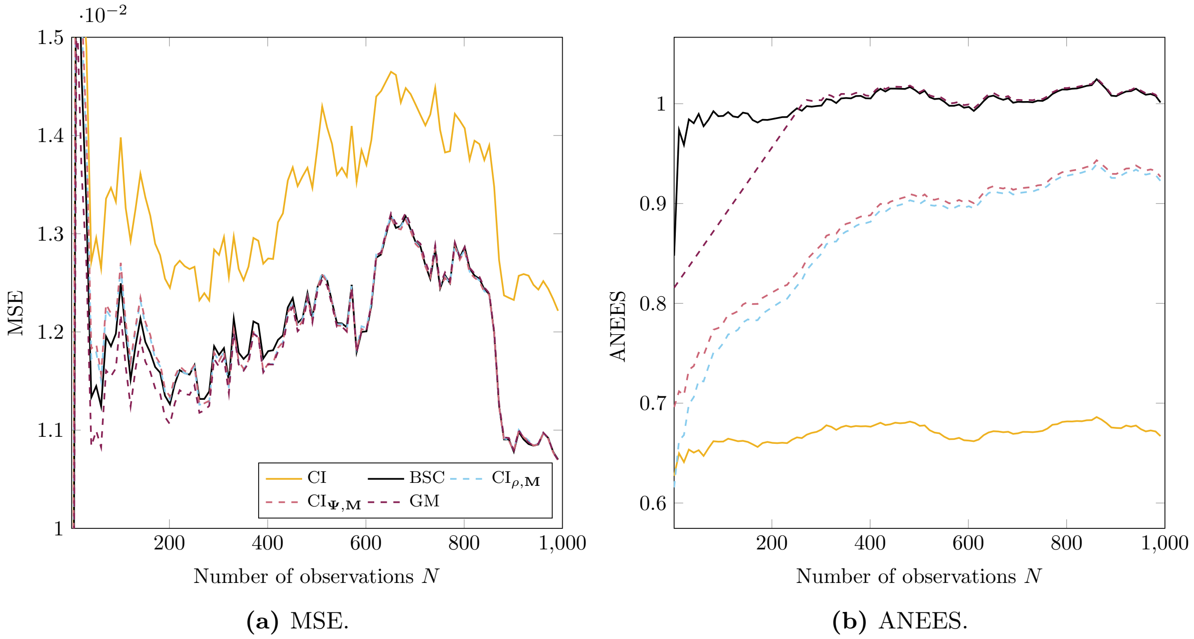
Node A sends only its first state $\underline{\hat{x}}_1^A$ and the belonging variance $\mathbf{P}_{11}^A$ to the fusion center, while node B sends the full state estimate $\underline{\hat{x}}^B$ and covariance $\mathbf{P}^B$.

6.4.1 Evaluation of the Simulation Approach

First, the simulation approach is evaluated where the correlation matrix is learned using several MCRs of the complete system, including the local estimators and the fusion step. The used setup is also proposed in [150], where the use of the sample correlation is evaluated. Here, the correlation is estimated using the particle approach from Chapter 5 and then the bounds proposed in 6.3.2 are used for the fusion. It is assumed that the employed local system and measurement models are unknown to the fusion center, but static. The correlation of the state estimates is estimated based on the output of the local estimators and the estimation error.

A Learning Correlations for Single Initial Covariance

First, the correlation between estimation errors is estimated by drawing several observations using only a single known initial covariance matrix $\mathbf{P}_0$. Therefore, the Kalman filters of the local



(a) MSE.

(b) ANEES.

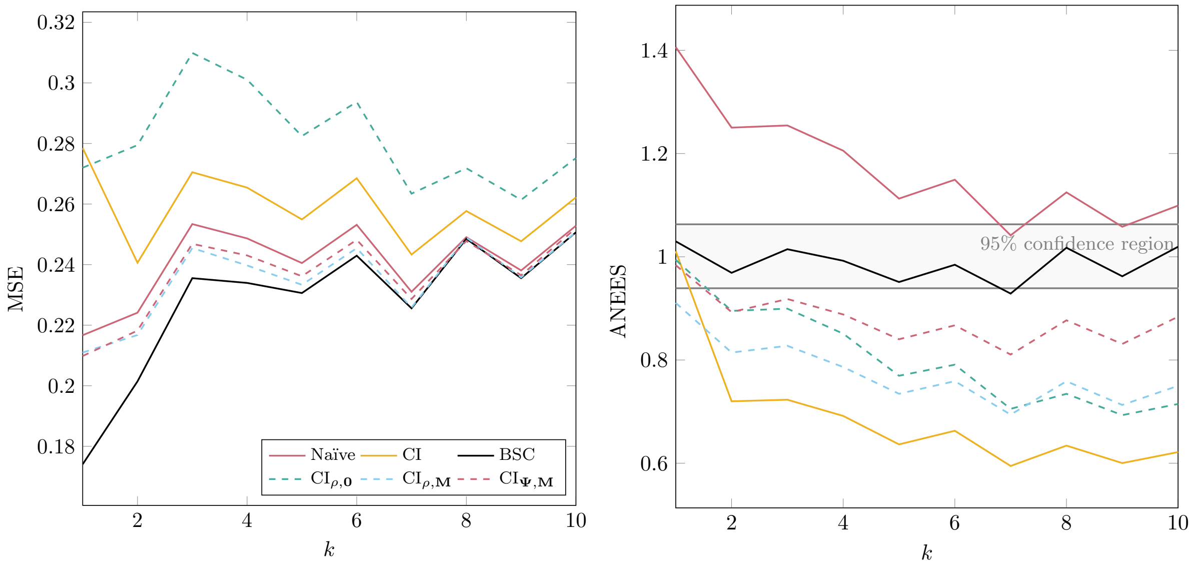
Figure 6.12: Comparison of fusion results using the simulation approach for learning of bounds for one initial covariance matrix using different fusion methods, 50 MCRs, moving average over the last 200 values.

sensor nodes are initialized with the same initial covariance matrix $\mathbf{P}_0 = [1, 0.5; 0.5, 1]$. As already discussed in (5.2), the joint covariance matrix is given by

$$\mathbf{J} = \begin{bmatrix} S^A & 0 \\ 0 & \mathbf{S}^B \end{bmatrix} \begin{bmatrix} 1 & \mathbf{\Lambda} \\ \mathbf{\Lambda}^\mathrm{T} & \mathbf{I} \end{bmatrix} \begin{bmatrix} S^A & 0 \\ 0 & \mathbf{S}^B \end{bmatrix}^\mathrm{T} ,$$

where S^A is the square-root of the variance $\mathbf{P}^A_{11}$ and $\mathbf{S}^B$ is the Cholesky decomposition of $\mathbf{P}^B$. The correlation matrix $\mathbf{\Lambda} = [\varrho^{1,1}, \varrho^{1,2}]$ contains two correlation coefficients that describe how the two state estimates are correlated. Figure 6.11 compares the estimated and the true correlation coefficients. Furthermore, the 95% confidence interval is plotted. While the uncertainty is large initially, the estimation improves with a growing number of observations and converges towards the actual value.

When no or only a few observations are available to the estimator, no knowledge about the fusion is available to the fusion center and, therefore, CI is used. The performance of the fusion with different fusion methods is shown in 6.12. Overall, all methods using learned partial knowledge show better performance than the standard CI. The GM approach converges very fast towards the optimal fusion result. The bounding methods using scalar values ($\text{CI}_{\rho,\mathbf{M}}$) and bounding with the scaling matrices $\mathbf{\Psi}$ ($\text{CI}_{\mathbf{\Psi},\mathbf{M}}$) also converge fast. However, their difference is insignificant, meaning that the uncertainty of the correlation estimate is not very elliptical and cannot be exploited. Furthermore, both bounds are still relatively conservative compared to the proposed GM approach, which is almost optimal.



(a) MSE.

(b) ANEES.

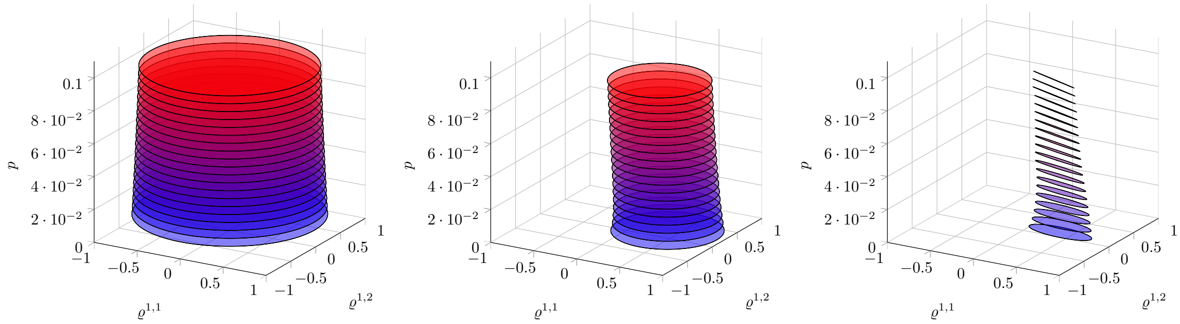
Figure 6.13: Comparison of fusion results using the analytic approach for learning of bounds for one initial covariance matrix using different fusion methods, 10000 MCRs.

6.4.2 Evaluation of the Analytic Approach

The following section evaluates the analytic approach using the previously proposed system design. However, it is now assumed that the fusion step does not occur after the first processing step but may occur randomly between $k = 1$ and $k = 10$. The set of cross-correlation matrices is reconstructed fully, using the square-root decomposition as proposed in Chapter 3, and then normalized. Then, the bounding of the set of correlation coefficients is executed using the methods proposed in section 6.3.1. This calculation of the bounding parameters using the set of correlation coefficients is done offline. Then the derived bounds are used in a simulation with 1000 MCRs, where the fusion step is chosen randomly.

A Learning Correlations for Single Initial Covariance

First, the evaluation is executed for a single initial covariance matrix $\mathbf{P}_0 = p[1, 0.5; 0.5, 1]$, where the scaling factor $p = 0.1$. Figure 6.12 shows the performance of the fusion using different methods depending on the time step when the fusion was executed. As before, the naïve fusion, CI and the optimal fusion using the known cross-covariance that is reconstructed during the simulation are compared as well to give a baseline for the performance. Figure 6.13a shows that the bounding with a scalar value and zero centering matrix ($\mathrm{CI}_{\rho,\mathbf{0}}$) performs the worst with CI following, which was surprising. The bounding with the scaling matrices $\mathrm{CI}_{\mathbf{\Psi},\mathbf{M}}$ performs best after the optimal fusion method, followed by the bound using the scalar value and centering matrix ($\mathrm{CI}_{\rho,\mathbf{M}}$), which also performed well. Furthermore, Figure 6.13b shows that the elliptical bound is the least conservative bounding method, followed by the two bounding methods with scalar values.



(a) Bounds for $\mathrm{CI}_{\rho,\mathbf{0}}$. **(b)** Bounds for $\mathrm{CI}_{\rho,\mathbf{M}}$. **(c)** Bound for $\mathrm{CI}_{\mathbf{\Psi},\mathbf{M}}$.

Figure 6.14: Learned bounding ellipses for different scaled initial covariance matrices over the scaling factor p.

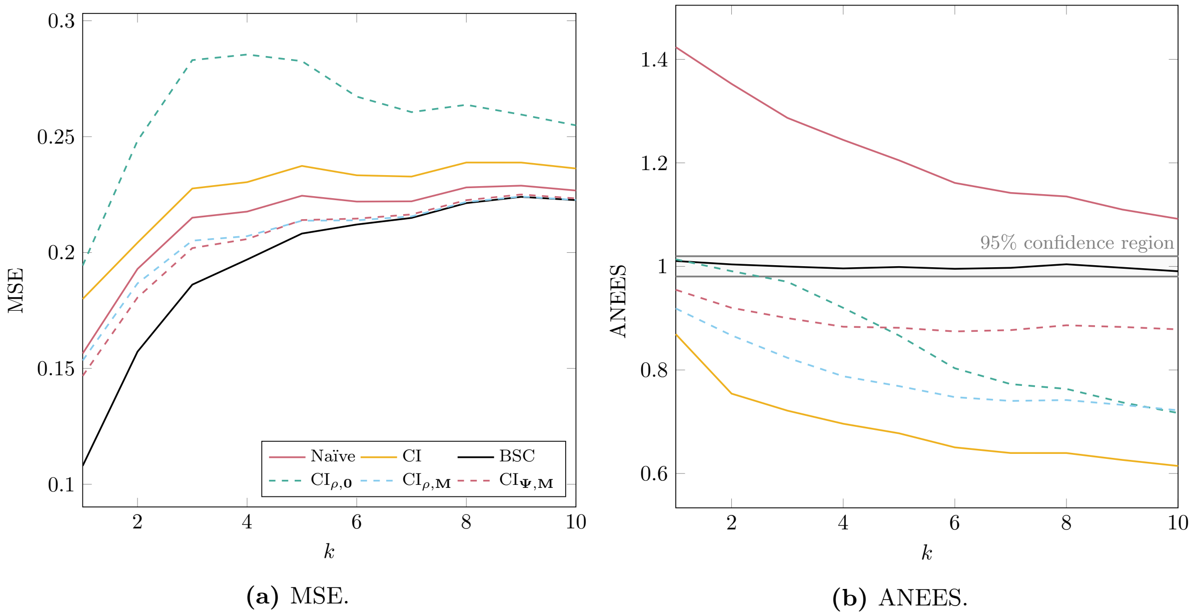
B Learning Correlations for Several Initial Covariances

Now, the evaluation is extended to the learning over several initial covariances. It is assumed that the scaling factor p is randomly chosen between $p_{\min} = 0.01$ and $p_{\max} = 0.1$. The cross-correlation is reconstructed fully for 20 equally distributed values between $p_{\min} = 0.01$ and $p_{\max} = 0.1$. These correlation sets are used to learn the bounding parameters with a specific scaling value p. The bounding ellipses or circles over the scaling parameter p for the different kinds of bounds are depicted in Figure 6.14. The bounds for scalar values ρ are relatively large, where the bounds for a zero centering matrix $\mathbf{M} = \mathbf{0}$ are the largest. In comparison, the bounding ellipses (see Figure 6.14c) are very small and cover a small area. Also, they do not violate the bounds of the correlation coefficients, unlike the bound with the scalar value and a nonzero centering matrix.

When executing the simulation, the current scaling factor of the initial covariance matrix is extracted. Then, the database is searched for bounds with the closest entries for the scaling value. Finally, the new bound is calculated by interpolating between the bounding parameters based on the distance to the current scaling factor. The results from 6.15 show again that the bounding with a scalar value and zero centering matrix ($\mathrm{CI}_{\rho,\mathbf{0}}$) performs the worst with CI following. The bounding with the scaling matrices $\mathrm{CI}_{\mathbf{\Psi},\mathbf{M}}$ performs best after the optimal fusion method. It is also the least conservative bounding technique, followed by the bound using a scalar value and centering matrix ($\mathrm{CI}_{\rho,\mathbf{M}}$), which also performs well. In general, the learned bounds are rather conservative because of the large spread of possible sets of correlation matrices. This shows that the partial knowledge improves the fusion, but the more information is available, the better the fusion result becomes.

6.5 Conclusions to Exploiting Partial Knowledge about Correlation

This chapter explored different methods to exploit partial knowledge consisting of sets of correlation coefficients. These learned sets were retrieved either from full or partial analytic reconstruction or estimated from simulation, and a suitable system design for both approaches was proposed.



(a) MSE.

(b) ANEES.

Figure 6.15: Comparison of fusion results using the analytic approach for learning of bounds for several scaled initial covariance matrices using different fusion methods, 10000 MCRs.

Two main strategies were introduced to account for these sets of correlation coefficients. First, suitable bounding methods were introduced. Two promising methods that achieve tight bounds were chosen from the current state of the art and applied to the considered problem. The necessary parameterization for the bounds was directly calculated from the sets of correlation coefficients. Since bounding methods tend to overestimate the uncertainty of the fused estimate, a GM approach was proposed that can use the sets of correlation coefficients directly.

Several interesting findings result from this chapter. First, the GM approach shows an almost optimal performance when the utilized set of correlation coefficients has a small uncertainty. It is also a suitable approach to directly use the estimated correlation using the proposed particle approach from Chapter 5. Proper bounds can be learned either by analytic or by simulation approach. However, when bounding sets of correlation coefficients, the natural constraints of the correlation can be violated when the bounded correlation coefficient also includes values that are not admissible. This violation of constraints is less likely to occur when using the bounding technique with scaling matrices $\mathbf{\Psi}$ compared to using a single scalar value ρ. This overbounding seems to affect the fusion result negatively and leads to overly conservative fusion results, but future research is necessary to investigate this behavior. The bounding technique with scaling matrices $\mathbf{\Psi}$ seems to have more advantages in the analytic approach than in the simulation approach because the bound on the estimated correlation coefficient is more a circle than an ellipse, limiting its advantage. Combinations of the simulation and the analytic approach are possible, e.g., one could simulate several possible sets of systems and then use the analytic approach on the resulting data. The proposed methods can be adapted to a wide range of systems with missing knowledge about correlated estimation errors, and future research should seek to apply these methods to practical applications.

CHAPTER 7

Conclusions and Future Research

Distributed estimation is an important task of many sensor networks. While the local processing of measurements allows for a decentralized network topology that leads to more robust, scalable and modular applications, many distributed estimation applications suffer from correlated estimation errors that prevent proper fusion of locally processed state estimates. Neglecting correlated estimation errors or assuming overly conservative bounds can have negative impacts on the quality of the fusion results.

7.1 Key Contributions

This thesis focused on the retrieval of full or partial knowledge about correlated estimation errors in distributed estimation. The key contributions of this thesis and their implications are discussed in the following.

Full and Partial Reconstruction of Cross-Covariances: This thesis investigated two previously proposed approaches for reconstructing cross-covariances using deterministic samples and square root decompositions of correlated noise covariances. Based on the square root decomposition technique, a recursive track-keeping method using a matrix including square root decompositions until a user-defined time horizon was proposed. The track-keeping of correlated estimation errors suffers from a growing number of noise covariances that need to be stored, processed, and communicated. To alleviate bandwidth restrictions, a moving horizon approach was proposed to keep track of a limited number of correlated noise terms. This partial reconstruction of cross-covariances is possible due to the bounding of the residual term. Both methods are suitable to optimally reconstruct cross-covariances in linear systems. The methods were extended to reconstruct cross-covariances in systems with heterogeneous state-space representations subject to linear transformations. It has been shown that the design of the systems has to be done with care to derive credible local trackers and reconstruct cross-covariances that can be used in the fusion step. A significant contribution is the generalization of the reconstruction to decentralized sensor networks that is valuable for many sensor networks. It was shown that track-keeping of correlated estimation errors outperforms many other fusion methods and leads to a high convergence rate towards a global consensus.

It was found that the square root decomposition is the most suitable approach for the distributed track-keeping in sensor networks as it is easy to implement and use. Furthermore, it is more straight-forward to discard old information that does not contribute to the cross-covariance as much as newer information using a sliding window approach. This technique offers a flexible trade-off between the communication requirements and the quality of the fusion result. The performance can be tuned by adjusting the user-defined time horizons for correlated common prior information and process noise, and correlated measurement information. On the other hand, the reconstruction using deterministic samples is more complicated but might be better suited for nonlinear transformations between different state-space representations of the local estimators. However, nonlinear transformations between state-space representations pose several challenges, and future research is necessary to identify common information and fuse estimates correctly.

Retrieving Partial Knowledge about Correlated Estimation Errors: In some systems, the reconstruction of cross-covariances is impossible, e.g., because the system parameters of the local estimators are unknown. This could be the case when the system runs third-party software or when the local computational power is not sufficient to run the proposed reconstruction algorithms. In this case, a simulation-based approach was proposed that simulates the complete system, including local estimators and the fusion step. This information is used to estimate the correlation between estimation errors. An essential aspect of this estimation is that the correlation coefficients have a specific range of values that depends on other correlation coefficients to ensure that the joint covariance matrix is positive semi-definite. Therefore, we proposed to use either the sample correlation coefficient that estimates correlations independent of other correlation coefficients or to use a particle filter approach to maximize the likelihood function of the correlation and that can take the whole joint covariance matrix into account. Both approaches show good performance, where the sample correlation is less computationally demanding, but the particle approach better handles the natural constraints of correlation coefficients.

Exploiting Partial Knowledge about Correlated Estimation Errors: The full or partial analytic reconstruction and the proposed estimation of correlation coefficients can be used as a toolbox to retrieve partial knowledge about correlated estimation errors from sensor networks. We proposed two possible methods to retrieve this partial knowledge: an analytic and a simulation approach. Cross-covariances for admissible models are fully or partially reconstructed and normalized to obtain the correlation matrix during the analytic approach. During the simulation approach, simulated data of the local estimation steps and the fusion is used to estimate the correlation from the estimation errors. The result is a set of admissible correlation coefficients that are exploited for the fusion step. Therefore, we proposed a suitable system design for the simulation and the analytic approach.

We proposed suitable bounding methods and a Gaussian mixture (GM) approach to exploit these sets of correlation coefficients. The GM approach can use the particles from the SIS approach directly and shows almost optimal fusion results when the uncertainty of the set of correlation coefficients is small. However, this approach needs further investigation if it is credible in all circumstances. Moreover, it should be investigated how to further process the mixture in the local estimators and derive the resulting implications for the next fusion step. We show a data-driven approach to identify the parameters for the bounding methods from either the simulation or the analytic approach and evaluated both approaches using a low dimensional example with only two correlation coefficients. The results show that it is beneficial to use the proposed methods to exploit learned partial knowledge during the fusion step. Some open questions remain for parameter identification, especially for higher-dimensional problems. We proposed learning for a limited set of system inputs by only considering the initial covariance for a linear estimator.

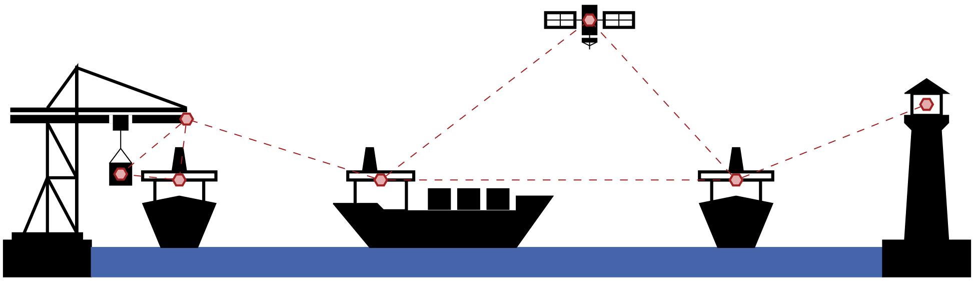


Figure 7.1: Example of a sensor network with different heterogeneous information sources, sensor nodes and connections between them in red.

However, many applications have arbitrary inputs, e.g., when the fusion result is used as feedback to the local estimators, and many systems also contain nonlinearities in system or measurement models. Therefore, further research is required to find if the proposed approach can still be used in these cases and how learning can be done efficiently.

7.2 Relevance

Fusing state estimates with correlated estimation errors is present in many research fields, such as target tracking or indoor localization. The fusion of state estimates is an essential part of many sensor networks and will even gain relevance in the coming years as more and more sensors and autonomous systems are employed. Because of the increased computational power, many systems will fulfill parts of their tasks autonomously but exchange local data with neighboring nodes to improve their local knowledge. This results in more decentralized systems containing correlated estimation errors that need to be addressed. An example for such a system is depicted in Figure 7.1. In maritime applications, vessels and ground stations exchange heterogeneous information, e.g., using the Automatic Identification System (AIS) to exchange static and dynamic information about the navigation of vessels, but also radar, active and passive sonar, or camera data [18]. This exchange of information is also subject to correlated estimation errors and challenging because of the decentralized nature of these systems. The methods from this thesis could be used by engineers working with or developing applications using distributed estimation and serve as building blocks to address correlated estimation errors. Depending on the application, the source of correlated estimation errors, and the provided bandwidth, users can choose a method to retrieve knowledge about correlated estimation errors either by reconstruction or by simulation and then use the proposed fusion methods to ensure credible fusion results.

7.3 Future Work

While this thesis answered several important research questions, it also shed light on other and sometimes entirely new essential aspects that need further investigation. Therefore, three follow-up research questions are listed here that we feel are worth considering in the future.

Cooperative Localization: Correlated estimation errors also appear in applications, where a team of mobile robots use relative measurements with respect to each other to update their position.

These relative measurements can be helpful in environments where there are few or sporadic landmarks [74]. Jointly estimating the pose of all team members increases the accuracy of the entire team. The sporadic access to precise localization information (e.g., landmarks) benefits the localization of all robots. This global performance increase is possible due to the coupling between local estimates created through the joint state estimation process. Unfortunately, robots share local information possibly several times, and the communication paths are hard to keep track of. However, precise estimation is only possible if correlations between local estimation errors are known. Existing approaches often use CI to account for unknown correlated estimation errors [20, 79, 88], or require certain assumptions on the network topology or the communication scheme [74]. However, the local estimates of each mobile robot are never fully correlated which makes CI not beneficial. Furthermore, using assumptions about the network or the communication is infeasible because requirements for communication cannot be met in all circumstances. Therefore, the square root decomposition technique could be applied to keep track of correlations without requirements for communication. This allows for fully decentralized estimation while optimally keeping track of all possible sources of correlated estimation errors. In addition, the sliding window approach to partially bound correlated estimation errors can help reduce bandwidth requirements.

Compression of Data to Reduce Bandwidth Requirements: Chapters 3 and 4 were concerned with the full and partial reconstruction of cross-covariances, where the partial reconstruction was motivated by the limited amount of communication bandwidth usually available to sensor networks. However, to reduce bandwidth requirements, data can be compressed, e.g., using quantization, to reduce the amount of additional information. Compression algorithms are often applied before transmitting data between sensor nodes [127] since data is digitally transmitted by a data package of a specific size. Therefore, data have a certain precision. This is an interesting aspect as this leads to imprecision in the reconstruction of cross-covariances, leading to underestimating correlated estimation errors. Moreover, to use the technique fully, it is necessary to investigate how compression influences the quality of the reconstruction. In [52, 53] quantization of data was investigated for CI, which could be a good starting point for further research.

Nonlinear Fusion of State Estimates: Especially tracking problems often feature local estimators employing nonlinear system or measurement models. Local estimation errors are often correlated due to common process noise. However, when the process noise model is nonlinear, every local tracker incorporates different process noise covariances depending on the current local state estimate. This complicates the identification of common process noise. Therefore, the reconstruction of correlated estimation errors is unclear. Yet, it is crucial to investigate how the local nonlinearities affect the correlation of estimation errors and how common information can be identified and reconstructed. Another problem are nonlinear transformations into local coordinate systems, as it is often the case in Heterogeneous Track-to-Track Fusion (HT2TF) problems. The result can be a joint probability density that is possibly non-Gaussian. Therefore nonlinear fusion methods are needed requiring further research. Tracking applications might also contain non-Gaussian local estimates, e.g., Gaussian mixtures. Last, there are applications where the state representation lies on special manifolds, e.g., for directional estimation. There, the fusion of state estimates is challenging because of correlated estimation errors that may lie on a special manifold [107] or at least need special treatment of correlated estimation errors based on the underlying manifold.

List of Figures

1.1 Experimental setup for acoustic bearings-only tracking. 2

1.2 Processing steps for distributed estimation in sensor networks. 3

2.1 Different sensor networks topologies. 11

2.2 Information graph for different fusion topologies, with and without feedback from the fusion center. 12

2.3 Information graph for hierarchical fusion with feedback. 13

2.4 Information graph for decentralized fusion topologies that are fully or only sparsely connected. 14

2.5 Comparison of different fusion methods, error ellipses and simulation. 16

2.6 Types of Correlation Uncertainty. 18

3.1 Comparison of different sampling schemes. 25

3.2 Transformation from a two-dimensional to a one-dimensional state space. 31

3.3 Problem sketch for a heterogeneous track-to-track fusion with three local filters estimating linear subsystems of the global state space. 33

3.4 Evaluation of the fusion results of the HT2TF example. 35

3.5 Separation of the global state estimate into the two overlapping subsystems. . . . 37

3.6 Fusion results for overlapping state estimates with estimation in subspace and estimation in full state space. 40

3.7 Two decentralized network topologies. 42

4.1 Comparison of error ellipses for several fusion methods. 50

4.2 Evaluation results for the centralized fusion of two sensor nodes. 53

4.3 Evaluation results for the decentralized fusion of two sensor nodes. 58

4.4 Visualization of a ring topology and evaluation results for consensus between estimates using different fusion methods. 60

4.5 Evaluation results for fusion in ring topology performing several consensus steps. . 61

5.1 Convex Shapes of different correlation matrices in $\mathbb{R}^2$. 65

5.2 Convex Shapes of different correlation matrices in $\mathbb{R}^3$. 66

5.3 Comparison of the Fisher z-transform with a simple Gaussian approximation. . . . 68

5.4 Example of Sampling a uniform distribution of a correlation matrix with two correlation coefficients. 71

5.5 Particle distribution of the proposed filter at different number of observations. . . 72

5.6 Comparison of the sample correlation with the proposed particle filter. 73

6.1 Workflow for learning sets of correlation coefficients for two sensor nodes A and B using the analytic approach. 76

6.2 Workflow for learning sets of correlation coefficients for two sensor nodes A and B using the simulation approach. 77

6.3 Comparison of constrainted CI methods. 79

6.4 Visualization of the GM approach. 82

6.5 Comparison of different weighting techniques for the GM approach. 83

6.6 Evaluation of the GM approach with different number of particles. 84

6.7 Evaluation of the GM approach using different calculation methods of the fusion gain. 85

6.8 Visualization of full and partial analytic calculation of the cross-correlation. 86

6.9 Fitting of bounding circles and ellipses for the analytic approach with full and partial reconstruction 87

6.10 Fitting of bounding circles and ellipses for the simulation approach. 90

6.11 Estimated correlation coefficients. 91

6.12 Evaluation results for simulation approach using one intial covariance. 92

6.13 Evaluation results for analytic approach using one initial covariance. 93

6.14 Learned bounding ellipses for different scaled initial covariance matrices. 94

6.15 Evaluation results for analytic approach using several scaled initial covariances. . . 95

7.1 Example of a sensor network with different heterogeneous information sources. . . 99

List of Tables

4.1 Number of additional data packages neccessary for the reconstruction using SqRD, for the evaluation example of centralized fusion with two sensor nodes. 54

4.2 Number of additional data packages neccessary for the reconstruction using SqRD, for the evaluation example of decentralized fusion with two sensor nodes. 59

4.3 Number of additional data packages neccessary for the reconstruction using SqRD, for evaluation example of consensus in a hierarchical network. 62

Bibliography

[1] J. Ajgl and M. Simandl. On linear estimation fusion under unknown correlations of estimator errors. *IFAC Proceedings Volumes*, 47:2364–2369, 2014.

[2] J. Ajgl, M. Šimandl, M. Reinhardt, B. Noack, and U. D. Hanebeck. Covariance Intersection in State Estimation of Dynamical Systems. In *Proceedings of the 17th International Conference on Information Fusion (Fusion 2014)*, Salamanca, Spain, July 2014.

[3] J. Ajgl and O. Straka. On Weak Points of the Ellipsoidal Intersection Fusion. In *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 28–33, 2017.

[4] J. Ajgl and O. Straka. Comparison of Fusions Under Unknown and Partially Known Correlations. In *IFAC-PapersOnLine, 7th IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys18)*, volume 51, pages 295–300, Groningen, The Netherlands, Aug 2018.

[5] J. Ajgl and O. Straka. Covariance Intersection in Track-to-Track Fusion: Comparison of Fusion Configurations. *IEEE Transactions On Industrial Informatics*, 14(3):1127–1136, Mar 2018.

[6] J. Ajgl and O. Straka. Rectification of Partitioned Covariance Intersection. In *2019 American Control Conference (ACC)*, pages 5786–5791, 2019.

[7] J. Ajgl and O. Straka. Lower Bounds In Estimation Fusion With Partial Knowledge of Correlations. In *Proceedings of the 2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2021)*, Karlsruhe, Sep 2021.

[8] J. Ajgl, O. Straka, and U. D. Hanebeck. Project Meeting, May 26. 2021.

[9] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, Feb 2002.

[10] Y. Bar-Shalom and K. Birmiwal. Consistency and robustness of PDAF for target tracking in cluttered environments. *Autom.*, 19:431–437, 1983.

[11] Y. Bar-Shalom and L. Campo. On the Track-to-Track Correlation Problem. *IEEE Transactions on Automatic Control*, 26(2):571–572, Apr. 1981.

[12] Y. Bar-Shalom and L. Campo. The Effect of the Common Process Noise on the Two-Sensor Fused-Track Covariance. In *IEEE Transactions on aerospace and electronic systems*, volume Vol. AES-22. No. 6, pages 803–805, 1986.

[13] Y. Bar-Shalom and X. Li. *Multitarget-Multisensor Tracking: Principles and Techniques.* YBS Publishing, Storrs, CT, 1995.

[14] Y. Bar-Shalom, X. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. Wiley, New York, NY, 01 2004.

[15] G. Battistelli and L. Chisci. Kullback-Leibler Average, Consensus on Probability Densities, and Distributed State Estimation with Guaranteed Stability. *Automatica*, 50(3):707–718, Mar. 2014.

[16] G. Battistelli, L. Chisci, N. Forti, G. Pelosi, and S. Selleri. Distributed Finite-Element Kalman Filter for Field Estimation. *IEEE Transactions on Automatic Control*, 62(7):3309–3322, Jul 2017.

[17] G. Battistelli, L. Chisci, G. Mugnai, A. Farina, and A. Graziano. Consensus-Based Linear and Nonlinear Filtering. *IEEE Transactions on Automatic Control*, 60(5):1410–1415, May 2015.

[18] K. Bereta, A. Millios, K. Chatzikokolakis, and D. Zissis. Monitoring Marine Protected Areas using Data Fusion and AI Techniques. *1st Maritime Situational Awareness Workshop (MSAW)*, pages 1–5, 2019.

[19] A. L. Bowley. The Standard Deviation of the Correlation Coefficient. *Journal of the American Statistical Association*, 23(161):31–34, 1928.

[20] L. C. Carrillo-Arce, E. D. Nerurkar, J. L. Gordillo, and S. I. Roumeliotis. Decentralized Multi-Robot Cooperative Localization Using Covariance Intersection. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1412–1417, Nov 2013.

[21] C. Carthel, S. Coraluppi, and P. Grignan. Multisensor Tracking and Fusion for Maritime Surveillance. In *2007 10th International Conference on Information Fusion*, pages 1–6, Jul 2007.

[22] F. S. Cattivelli and A. H. Sayed. Diffusion Strategies for Distributed Kalman Filtering and Smoothing. *IEEE Transactions on Automatic Control*, 55(9):2069–2084, Sep 2010.

[23] K.-C. Chang, R. K. Saha, and Y. Bar-Shalom. On Optimal Track-to-Track Fusion. *IEEE Transactions on Aerospace and Electronic Systems*, 33(4):1271–1276, Oct. 1997.

[24] B. Chen, G. Hu, D. W. C. Ho, and L. Yu. A New Approach to Linear/Nonlinear Distributed Fusion Estimation Problem. *CoRR*, abs/1708.08583, 2017.

[25] H. Chen and Y. Bar-Shalom. Track Association and Fusion with Heterogeneous Local Trackers. In *2007 46th IEEE Conference on Decision and Control*, pages 2675–2680, Dec 2007.

[26] L. Chen, P. Arambel, and R. Mehra. Fusion Under Unknown Correlation - Covariance Intersection as a Special Case. In *Proceedings of the Fifth International Conference on Information Fusion (Fusion 2002)*, volume 2, pages 905–912, 2002.

[27] L. Chen, P. O. Arambel, and R. K. Mehra. Fusion under Unknown Correlation: Covariance Intersection Revisited. *IEEE Transactions on Automatic Control*, 47(11):1879–1882, Nov. 2002.

[28] C.-Y. Chong. Hierarchical Estimation. In *MIT/ONR Workshop on C3 Systems*, pages 205–220, Monterey, California, USA, 1979.

[29] C.-Y. Chong, K.-C. Chang, and S. Mori. Distributed Tracking in Distributed Sensor Networks. In *Proceedings of the 1986 American Control Conference (ACC 1986)*, pages 1863–1868, Seattle, Washington, USA, 1986.

[30] C.-Y. Chong, K.-C. Chang, and S. Mori. Fundamentals of Distributed Estimation. In D. L. Hall, C.-Y. Chong, J. Llinas, and M. E. Liggins II, editors, *Distributed Data Fusion for Network-Centric Operations*, The Electrical Engineering and Applied Signal Processing Series, pages 95–124. CRC Press, Boca Raton, Florida, USA, 2013.

[31] C.-Y. Chong, K.-C. Chang, and S. Mori. Comparison of Optimal Distributed Estimation and Consensus Filtering. In *Proceedings of the 19th International Conference on Information Fusion (Fusion 2016)*, pages 1034–1041, 2016.

[32] C.-Y. Chong, K.-C. Chang, and S. Mori. A Review of Forty Years of Distributed Estimation. In *Proceedings of the 21st International Conference on Information Fusion (Fusion 2018)*, pages 1–8, Cambridge, United Kingdom, July 2018.

[33] C.-Y. Chong and S. Kumar. Sensor Networks: Evolution, Opportunities, and Challenges. *Proceedings of the IEEE*, 91(8):1247–1256, Aug 2003.

[34] C.-Y. Chong and S. Mori. Convex Combination and Covariance Intersection Algorithms in Distributed Fusion. In *Proceedings of the 4th International Conference on Information Fusion (Fusion 2001)*, Montréal, Québec, Canada, Aug. 2001.

[35] C.-Y. Chong, S. Mori, and K.-C. Chang. Information Fusion in Distributed Sensor Networks. In *1985 American Control Conference*, pages 830–835, Jun 1985.

[36] C.-Y. Chong, E. Tse, and S. Mori. Distributed Estimation in Networks. In *1983 American Control Conference*, pages 294–300, Jun 1983.

[37] J. Čurn, D. Marinescu, N. O'Hara, and V. Cahill. Data Incest in Cooperative Localisation with the Common Past-Invariant Ensemble Kalman Filter. In *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, pages 68–76, Istanbul, Turkey, July 2013.

[38] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[39] D. Ding, Z. Wang, and B. Shen. Recent Advances on Distributed Filtering for Stochastic Systems over Sensor Networks. *International Journal of General Systems*, 43(3-4):372–386, 2014.

[40] A. Doucet, N. de Freitas, and N. J. Gordon. An introduction to sequential monte carlo methods. In *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, USA, 2001.

[41] M. Erdelj, N. Mitton, and E. Natalizio. Applications of Industrial Wireless Sensor Networks. In G. Hancke, editor, *Industrial Wireless Sensor Networks*, volume 20132544. CRC Press, Apr. 2013.

[42] L. Fahrmeir, T. Kneib, S. Lang, and B. Marx. *Regression: Models, Methods and Applications.* Springer-Verlag Berlin Heidelberg, Jan 2013.

[43] M. Farina, G. Ferrari Trecate, and R. Scattolini. Moving-Horizon Partition-based State Estimation of Large-Scale Systems. *Automatica*, 46:910–918, May 2010.

[44] R. A. Fisher. On the "Probable Error" of a Coefficient of Correlation Deduced from a Small Sample. *Metron*, 1:3–32, 1921.

[45] P. J. Forrester and J. Zhang. Parametrising Correlation Matrices. *Journal of Multivariate Analysis*, 178(C), 2020.

[46] R. Forsling, Z. Sjanic, F. Gustafsson, and G. Hendeby. Communication Efficient Decentralized Track Fusion Using Selective Information Extraction. In *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pages 1–8, Jul 2020.

[47] W. Förstner and B. Moonen. *A Metric for Covariance Matrices*, pages 299–309. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[48] B. K. Fosdick and A. E. Raftery. Estimating the Correlation in Bivariate Normal Data With Known Variances and Small Sample Sizes. *The American Statistician*, 66(1):34–41, 2012.

[49] M. Friendly, G. Monette, and J. Fox. Elliptical Insights: Understanding Statistical Methods through Elliptical Geometry. *Statistical Science*, 28(1):1–39, 2013.

[50] D. Frisch and U. D. Hanebeck. Deterministic Gaussian Sampling With Generalized Fibonacci Grids. In *Proceedings of the 24rd International Conference on Information Fusion (Fusion 2021)*, South Africa, Nov. 2021.

[51] D. Frisch and U. D. Hanebeck. Gaussian Mixture Estimation from Weighted Samples. In *Proceedings of the 2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2021)*, Karlsruhe, Sept. 2021.

[52] C. Funk, B. Noack, and U. D. Hanebeck. Conservative Quantization of Fast Covariance Intersection. In *Proceedings of the 2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2020)*, Virtual, Sept. 2020.

[53] C. Funk, B. Noack, and U. D. Hanebeck. Conservative Quantization of Covariance Matrices with Applications to Decentralized Information Fusion. *Sensors*, Apr. 2021.

[54] N. Gordon, D. Salmond, and A. Smith. Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, page 107. Institution of Engineering and Technology (IET), 1993.

[55] F. Govaers and W. Koch. On the Globalized Likelihood Function for Exact Track-to-Track Fusion at Arbitrary Instants of Time. In *14th International Conference on Information Fusion*, pages 1–5, July 2011.

[56] F. Govaers and W. Koch. An Exact Solution to Track-to-track Fusion at Arbitrary Communication Rates. *IEEE Transactions on Aerospace and Electronic Systems*, 48(3):2718–2729, July 2012.

[57] S. Grime and H. F. Durrant-Whyte. Data Fusion in Decentralized Sensor Networks. *Control Engineering Practice*, 2(5):849–863, Oct. 1994.

[58] D. L. Hall, C.-Y. Chong, J. Llinas, and M. E. Liggins II, editors. *Distributed Data Fusion for Network-Centric Operations*. The Electrical Engineering and Applied Signal Processing Series. CRC Press, Boca Raton, Florida, USA, 2013.

[59] U. D. Hanebeck, K. Briechle, and J. Horn. A Tight Bound for the Joint Covariance of Two Random Vectors with Unknown but Constrained Cross–Correlation. In *Proceedings of the 2001 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2001)*, pages 85–90, Baden–Baden, Germany, Aug. 2001.

[60] H. Hashemipour, S. Roy, and A. Laub. Decentralized Structures for Parallel Kalman Filtering. *IEEE Transactions on Automatic Control*, 33(1):88–94, 1988.

[61] S. He, H.-S. Shin, S. Xu, and A. Tsourdos. Distributed estimation over a low-cost sensor network: A Review of state-of-the-art. *Information Fusion*, 54:21–43, 2020.

[62] J. Hu, L. Xie, and C. Zhang. Diffusion Kalman Filtering Based on Covariance Intersection. *IEEE Transactions on Signal Processing*, 60:891–902, Feb 2012.

[63] M. Ikeda, D. D. Siljak, and D. E. White. Decentralized Control with Overlapping Information Sets. *Journal of Optimization Theory and Applications*, 34:279–310, 1981.

[64] H. Joe. Generating Random Correlation Matrices Based on Partial Correlations. *Journal of Multivariate Analysis*, 97:2177–2189, Feb 2006.

[65] S. Julier. The Spherical Simplex Unscented Transformation. *Proceedings of the 2003 American Control Conference, 2003.*, 3:2430–2434 vol.3, 2003.

[66] S. J. Julier. Estimating and Exploiting the Degree of Independent Information in Distributed Data Fusion. In *Proceedings of the 12th International Conference on Information Fusion (Fusion 2009)*, Seattle, Washington, USA, July 2009.

[67] S. J. Julier and J. K. Uhlmann. A Non-divergent Estimation Algorithm in the Presence of Unknown Correlations. In *Proceedings of the IEEE American Control Conference (ACC 1997)*, volume 4, pages 2369–2373, Albuquerque, New Mexico, USA, June 1997.

[68] S. J. Julier and J. K. Uhlmann. Unscented Filtering and Nonlinear Estimation. In *Proceedings of the IEEE*, volume 92, pages 401–422, Mar 2004.

[69] S. J. Julier and J. K. Uhlmann. *Handbook of Multisensor Data Fusion: Theory and Practice*, chapter General Decentralized Data Fusion with Covariance Intersection, pages 319–343. CRC Press, Boca Raton, Florida, USA, 2nd ed. edition, 2009.

[70] P. Kalata and S. L. Fagin. Kalman Filtering: One Form Fits All - An Innovative, Pure Square Root Process. In *1990 American Control Conference*, pages 2959–2964, 1990.

[71] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, Mar 1960.

[72] S. K. Khaitan and J. D. McCalley. Design Techniques and Applications of Cyberphysical Systems: A Survey. *IEEE Systems Journal*, 9(2):350–365, Jun 2015.

[73] U. A. Khan and J. M. F. Moura. Distributing the Kalman Filter for Large-Scale Systems. *IEEE Transactions on Signal Processing*, 56(10):4919–4935, Oct 2008.

[74] S. S. Kia, S. Rounds, and S. Martinez. Cooperative Localization for Mobile Agents: A Recursive Decentralized Algorithm Based on Kalman-Filter Decoupling. *IEEE Control Systems Magazine*, 36(2):86–101, 2016.

[75] V. Klumpp and U. D. Hanebeck. Bayesian Estimation with Uncertain Parameters of Probability Density Functions. In *Proceedings of the 12th International Conference on Information Fusion (Fusion 2009)*, Seattle, Washington, USA, July 2009.

[76] W. Koch. On Optimal Distributed Kalman Filtering and Retrodiction at Arbitrary Communication Rates for Maneuvering Targets. In *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 457–462, Aug 2008.

[77] W. Koch. *Tracking and Sensor Data Fusion: Methodological Framework and Selected Applications*. Springer Verlag, Berlin Heidelberg, Germany, Oct 2014.

[78] P. Kumar and E. A. Yildirim. Minimum-Volume Enclosing Ellipsoids and Core Sets. *Journal of Optimization Theory and Applications*, 126(1):1–21, 2005.

[79] J. Lai, Y. Zhou, J. Lin, Y. Cong, and J. Yang. Cooperative Localization Based on Efficient Covariance Intersection. *IEEE Communications Letters*, 23(5):871–874, May 2019.

[80] M. Laurent and S. Poljak. On a Positive Semidefinite Relaxation of the Cut Polytope. *Linear Algebra and its Applications*, 223-224:439–461, 1995. Honoring Miroslav Fiedler and Vlastimil Ptak.

[81] T. Lefebvre, H. Bruyninckx, and J. de Schutter. *Nonlinear Kalman Filtering for Force-Controlled Robot Tasks*, volume 19 of *Springer Tracts in Advanced Robotics*. Springer, 2005.

[82] X. Li, Y. Zhu, J. Wang, and C. Han. Optimal Linear Estimation Fusion .I. Unified Fusion Rules. *IEEE Transactions on Information Theory*, 49(9):2192–2208, Sep 2003.

[83] X. R. Li, Z. Zhao, and V. P. Jilkov. Practical Measures and Test for Credibility of an Estimator. In *Proceedings of Workshop on Estimation, Tracking, and Fusion - A Tribute to Yaakov Bar-Shalom*, 2001.

[84] Z. Li, K. You, and S. Song. Batch CI-Based Kalman Smoother for PM2.5 Source Localization. In *2020 IEEE 16th International Conference on Control Automation (ICCA)*, pages 295–300, Oct 2020.

[85] Z. Li, K. You, and S. Song. Cooperative Field Prediction and Smoothing via Covariance Intersection. *IEEE Transactions on Signal Processing*, 69:797–808, 2021.

[86] M. Liggins, C.-Y. Chong, I. Kadar, M. Alford, V. Vannicola, and S. Thomopoulos. Distributed fusion architectures and algorithms for target tracking. *Proceedings of the IEEE*, 85(1):95–107, Jan 1997.

[87] J. Liu, Y. Liu, K. Dong, Z. Ding, and Y. He. A Novel Distributed State Estimation Algorithm with Consensus Strategy. *Sensors*, 19(9):1–29, 2019.

[88] I. Loefgren, N. Ahmed, E. Frew, C. Heckman, and S. Humbert. Scalable Event-Triggered Data Fusion for Autonomous Cooperative Swarm Localization. In *2019 22th International Conference on Information Fusion (FUSION)*, pages 1–8, 2019.

[89] D. Luengo, L. Martino, M. F. Bugallo, V. Elvira, and S. Särkkä. A Survey of Monte Carlo Methods for Parameter Estimation. *EURASIP Journal on Advances in Signal Processing*, 2020:1–62, 2020.

[90] M. Mallick, K.-C. Chang, S. Arulampalam, and Y. Yan. Heterogeneous Track-to-Track Fusion in 3-D Using IRST Sensor and Air MTI Radar. *IEEE Transactions on Aerospace and Electronic Systems*, 55(6):3062–3079, Dec 2019.

[91] M. Mallick, K.-C. Chang, S. Arulampalam, Y. Yan, and B. La Scala. Heterogeneous Track-to-Track Fusion in 2D Using Sonar and Radar Sensors. In *2019 22th International Conference on Information Fusion (FUSION)*, pages 1–8, Jul 2019.

[92] M. Mallick, B. La Scala, B. Ristic, T. Kirubarajan, and J. Hill. Comparison of Filtering Algorithms for Ground Target Tracking Using Space-based GMTI Radar. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 1672–1679, July 2015.

[93] N. Moshtagh. Minimum Volume Enclosing Ellipsoid, 8. Nov 2021. In *MATLAB Central File Exchange* [Online]. Available: `https://www.mathworks.com/matlabcentral/fileexchange/9542-minimum-volume-enclosing-ellipsoid`

[94] A. G. O. Mutambara. *Decentralized Estimation and Control for Multisensor Systems.* CRC Press, Boca Raton, Florida, USA, 1998.

[95] A. G. O. Mutambara and H. F. Durrant-Whyte. Nonlinear Information Space: A Practical Basis for Decentralization. In P. S. Schenker, editor, *Proceedings of SPIE's International Symposium on Photonics for Industrial Applications, Sensor Fusion VII*, volume 2355, pages 97–105. SPIE, 1994.

[96] W. Niehsen. Information Fusion based on Fast Covariance Intersection Filtering. In *Proceedings of the 5th International Conference on Information Fusion (Fusion 2002)*, pages 901–904 vol.2, Annapolis, Maryland, USA, July 2002.

[97] B. Noack, M. Baum, and U. D. Hanebeck. Covariance Intersection in Nonlinear Estimation Based on Pseudo Gaussian Densities. In *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)*, Chicago, Illinois, USA, July 2011.

[98] B. Noack, D. Lyons, M. Nagel, and U. D. Hanebeck. Nonlinear Information Filtering for Distributed Multisensor Data Fusion. In *Proceedings of the 2011 American Control Conference (ACC 2011)*, San Francisco, California, USA, June 2011.

[99] B. Noack, J. Sijs, and U. D. Hanebeck. Fusion Strategies for Unequal State Vectors in Distributed Kalman Filtering. In *Proceedings of the 19th IFAC World Congress (IFAC 2014)*, Cape Town, South Africa, Aug. 2014.

[100] B. Noack, J. Sijs, and U. D. Hanebeck. Algebraic Analysis of Data Fusion with Ellipsoidal Intersection. In *Proceedings of the 2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2016)*, Baden-Baden, Germany, Sept. 2016.

[101] B. Noack, J. Sijs, and U. D. Hanebeck. Inverse Covariance Intersection: New Insights and Properties. In *Proceedings of the 20th International Conference on Information Fusion (Fusion 2017)*, Xi'an, China, July 2017.

[102] B. Noack, J. Sijs, M. Reinhardt, and U. D. Hanebeck. Treatment of Dependent Information in Multisensor Kalman Filtering and Data Fusion. In H. Fourati, editor, *Multisensor Data Fusion: From Algorithms and Architectural Design to Applications*, pages 169–192. CRC Press, Aug. 2015.

[103] B. Noack, J. Sijs, M. Reinhardt, and U. D. Hanebeck. Decentralized Data Fusion with Inverse Covariance Intersection. *Automatica*, 79:35–41, May 2017.

[104] R. Olfati-Saber. Distributed Kalman Filter with Embedded Consensus Filters. In *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC 2005)*, pages 8179–8184, Sevilla, Spain, Dec. 2005.

[105] R. Olfati-Saber. Distributed Kalman Filtering for Sensor Networks. In *Proceedings of the 46th IEEE Conference on Decision and Control (CDC 2007)*, pages 5492–5498, New Orleans, Louisiana, USA, Dec. 2007.

[106] U. Orguner. Approximate Analytical Solutions for the Weight Optimization Problems of CI and ICI. In *Proceedings of the IEEE ISIF Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF 2017)*, Bonn, Germany, Oct. 2017.

[107] F. Pfaff, K. Li, and U. D. Hanebeck. Estimating Correlated Angles Using the Hypertoroidal Grid Filter. In *Proceedings of the 2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2020)*, Virtual, Sept. 2020.

[108] S. Reece and S. Roberts. Robust, Low-Bandwidth, Multi-Vehicle Mapping. In *2005 7th International Conference on Information Fusion*, volume 2, page 8, 2005.

[109] M. Reinhardt. *Linear Estimation in Interconnected Sensor Systems with Information Constraints*. Dissertation, Karlsruhe Institute of Technology (KIT), Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Referent: U. D. Hanebeck, Korreferent: S. R. Kulkarni, Karlsruhe Series on Intelligent Sensor-Actuator-Systems 16, 2014.

[110] M. Reinhardt, S. Kulkarni, and U. D. Hanebeck. Generalized Covariance Intersection based on Noise Decomposition. In *Proceedings of the 2014 IEEE International Conference on Multisensor Fusion and Information Integration (MFI 2014)*, Beijing, China, Sept. 2014.

[111] M. Reinhardt, B. Noack, P. O. Arambel, and U. D. Hanebeck. Minimum Covariance Bounds for the Fusion under Unknown Correlations. *IEEE Signal Processing Letters*, 22(9):1210–1214, Sept. 2015.

[112] M. Reinhardt, B. Noack, M. Baum, and U. D. Hanebeck. Analysis of Set-theoretic and Stochastic Models for Fusion under Unknown Correlations. In *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)*, Chicago, Illinois, USA, July 2011.

[113] M. Reinhardt, B. Noack, and U. D. Hanebeck. Closed-form Optimization of Covariance Intersection for Low-dimensional Matrices. In *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Singapore, July 2012.

[114] M. Reinhardt, B. Noack, and U. D. Hanebeck. The Hypothesizing Distributed Kalman Filter. In *Proceedings of the 2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2012)*, Hamburg, Germany, Sept. 2012.

[115] M. Reinhardt, B. Noack, and U. D. Hanebeck. Reconstruction of Joint Covariance Matrices in Networked Linear Systems. In *Proceedings of the 48th Annual Conference on Information Sciences and Systems (CISS 2014)*, Princeton, New Jersey, USA, Mar. 2014.

[116] B. Ristic and S. Arulampalam. *Beyond the Kalman filter : Particle Filters for Tracking Applications.* Artech House, Boston, MA, 2004.

[117] P. J. Rousseeuw and G. Molenberghs. The Shape of Correlation Matrices. *The American Statistician*, 48(4):276–279, 1994.

[118] V. Shin, Y. Lee, and T.-S. Choi. Generalized Millman's Formula and Its Application for Estimation Problems. *Signal Process.*, 86(2):257–266, Feb. 2006.

[119] J. Sijs, U. Hanebeck, and B. Noack. An Empirical Method to Fuse Partially Overlapping State Vectors for Distributed State Estimation. In *2013 European Control Conference (ECC)*, pages 1615–1620, Jul 2013.

[120] J. Sijs and M. Lazar. State-fusion with Unknown Correlation: Ellipsoidal Intersection. *Automatica*, 48(8):1874–1878, Aug. 2012.

[121] D. Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches.* John Wiley & Sons, Hoboken, NJ, 2006.

[122] J. L. Speyer. Computation and Transmission Requirements for a Decentralized Linear-Quadratic-Gaussian Control Problem. *IEEE Transactions on Automatic Control*, 24(2):266–269, Apr. 1979.

[123] S. S. Stankovic, M. S. Stankovic, and D. M. Stipanovic. Consensus Based Overlapping Decentralized Estimator. *IEEE Transactions on Automatic Control*, 54(2):410–415, Feb 2009.

[124] J. Steinbring. Nonlinear estimation toolbox. [Online]. Available: `https://bitbucket.org/nonlinearestimation/toolbox`

[125] J. Steinbring, B. Noack, M. Reinhardt, and U. D. Hanebeck. Optimal Sample-Based Fusion for Distributed State Estimation. In *Proceedings of the 19th International Conference on Information Fusion (Fusion 2016)*, Heidelberg, Germany, July 2016.

[126] P. Sun and R. M. Freund. Computation of Minimum-Volume Covering Ellipsoids. *Operations Research*, 52(5):690–706, 2004.

[127] S. Sun, H. Lin, J. Ma, and X. Li. Multi-sensor Distributed Fusion Estimation with Applications in Networked Systems: A Review Paper. *Information Fusion*, 38:122–134, 2017.

[128] R. Swinbank and R. James Purser. Fibonacci Grids: A Novel Approach to Global Modelling. *Quarterly Journal of the Royal Meteorological Society*, 132(619):1769–1793, 2006.

[129] X. Tian and Y. Bar-Shalom. Exact Algorithms for Four Track-to-Track Fusion Configurations: All You Wanted to Know but Were Afraid to Ask. In *2009 12th International Conference on Information Fusion*, pages 537–544, Jul 2009.

[130] X. Tian and Y. Bar-Shalom. The Optimal Algorithm for Asynchronous Track-to-Track Fusion. In *Defense + Commercial Sensing*, 2010.

[131] X. Tian, Y. Bar-Shalom, T. Yuan, E. Blasch, K. Pham, and G. Chen. A Generalized Information Matrix Fusion based Heterogeneous Track-to-Track Fusion Algorithm. In I. Kadar, editor, *Signal Processing, Sensor Fusion, and Target Recognition XX*, volume 8050, pages 469 – 476. International Society for Optics and Photonics, SPIE, 2011.

[132] J. K. Uhlmann. General Data Fusion for Estimates with Unknown Cross Covariances. In I. Kadar and V. Libby, editors, *Signal Processing, Sensor Fusion, and Target Recognition V*, volume 2755, pages 536 – 547. International Society for Optics and Photonics, SPIE, 1996.

[133] J. K. Uhlmann. Covariance Consistency Methods for Fault-Tolerant Distributed Data Fusion. *Information Fusion*, 4(3):201–215, 2003.

[134] Z. Wu, Q. Cai, and M. Fu. Covariance Intersection for Partially Correlated Random Vectors. *IEEE Transactions on Automatic Control*, 63(3):619–629, 2018.

[135] K. Yang, Y. Bar-Shalom, and K.-C. Chang. Information Matrix Fusion for Nonlinear, Asynchronous and Heterogeneous Systems. In *2019 22th International Conference on Information Fusion (FUSION)*, pages 1–6, Jul 2019.

[136] K. Yang, Y. Bar-Shalom, and P. Willett. The Cross-Covariance for Heterogeneous Track-to-Track Fusion. In I. Kadar, E. P. Blasch, and L. L. Grewe, editors, *Signal Processing, Sensor/Information Fusion, and Target Recognition XXVIII*, volume 11018, pages 42 – 51. International Society for Optics and Photonics, SPIE, 2019.

[137] T. Yuan, Y. Bar-Shalom, and X. Tian. Heterogeneous Track-to-Track Fusion. In *14th International Conference on Information Fusion*, pages 1–8, Jul 2011.

[138] T. Yuan, Y. bar shalom, and X. Tian. Heterogeneous track-to-track fusion. *Fusion 2011 - 14th International Conference on Information Fusion*, 6:131–149, Jan 2011.

[139] M. Zarei-Jalalabadi, S. M. B. Malaek, and S. S. Kia. A Track-to-Track Fusion Method for Tracks With Unknown Correlations. *IEEE Control Systems Letter*, 2(2):189–194, Apr. 2018.

Supervised Student Theses

For privacy reasons, the name of the student is omitted in each entry.

[140] *Decentralized Target Tracking Using Acoustic Localization.* Master Thesis, Karlsruhe Institute of Technology (KIT), Nov. 2018.

[141] *Decentralized Fusion in Networks with Arbitrary Topology.* Master Thesis, Karlsruhe Institute of Technology (KIT), Mar. 2019.

[142] *UWB-Inertial Sensor Fusion for Indoor Positioning.* Master Thesis, Karlsruhe Institute of Technology (KIT), Nov. 2019.

Own Publications

[143] K. Li, D. Frisch, S. Radtke, B. Noack, and U. D. Hanebeck. Wavefront Orientation Estimation Based on Progressive Bingham Filtering. In *Proceedings of the IEEE ISIF Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF 2018)*, Oct. 2018.

[144] B. Noack, C. Funk, S. Radtke, and U. D. Hanebeck. State Estimation with Event-Based Inputs Using Stochastic Triggers. In *Proceedings of the 1st Virtual IFAC World Congress (IFAC-V 2020)*, July 2020.

[145] S. Radtke, K. Li, B. Noack, and U. D. Hanebeck. Comparative Study of Track-to-Track Fusion Methods for Cooperative Tracking with Bearings-only Measurements. In *Proceedings of the 2019 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2019)*, Taipei, Republic of China, May 2019.

[146] S. Radtke, B. Noack, and U. D. Hanebeck. Distributed Estimation using Square Root Decompositions of Dependent Information. In *Proceedings of the 22nd International Conference on Information Fusion (Fusion 2019)*, Ottawa, Canada, July 2019.

[147] S. Radtke, B. Noack, and U. D. Hanebeck. Distributed Estimation with Partially Overlapping States based on Deterministic Sample-based Fusion. In *Proceedings of the 2019 European Control Conference (ECC 2019)*, Naples, Italy, June 2019.

[148] S. Radtke, B. Noack, and U. D. Hanebeck. Fully Decentralized Estimation Using Square-Root Decompositions. In *Proceedings of the 23rd International Conference on Information Fusion (Fusion 2020)*, Virtual, July 2020.

[149] S. Radtke, B. Noack, and U. D. Hanebeck. Fully Decentralized Estimation Using Square-Root Decompositions. *Journal of Advances in Information Fusion*, 16(1):3–16, June 2021.

[150] S. Radtke, J. Ajgl, O. Straka, and U. D. Hanebeck. Learning and Exploiting Partial Knowledge in Distributed Estimation. In *Proceedings of the 2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2021)*, Karlsruhe, Sept. 2021.

[151] S. Radtke, B. Noack, and U. D. Hanebeck. Reconstruction of Cross-Correlations between Heterogeneous Trackers Using Deterministic Samples. In *Proceedings of the 1st Virtual IFAC World Congress (IFAC-V 2020)*, July 2020.

[152] S. Radtke, B. Noack, U. D. Hanebeck, and O. Straka. Reconstruction of Cross-Correlations with Constant Number of Deterministic Samples. In *Proceedings of the 21st International Conference on Information Fusion (Fusion 2018)*, Cambridge, United Kingdom, July 2018.