



CRANFIELD UNIVERSITY

Shaoming He

**Distributed Target Tracking Over a
Low-Cost Sensor Network**

School of Aerospace, Transport and Manufacturing

PhD Thesis

Cranfield University

School of Aerospace, Transport and Manufacturing

PhD Thesis

Academic Year 2016-2019

Shaoming He

Distributed Target Tracking Over a Low-Cost Sensor
Network

Supervisors:	Prof. Hyo-Sang Shin
	Prof. Antonios Tsourdos

September 2019

This thesis is submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy.

©Cranfield University 2019. All rights reserved. No part of this publication may
be reproduced without the written permission of the copyright owner.

Abstract

Proliferation of low-cost, lightweight, and power efficient sensors and advances in networked systems enable the employment of multiple sensors. Distributed estimation provides a scalable and fault-robust fusion framework with a peer-to-peer communication architecture. This thesis investigates the problem of distributed target(s) tracking over a low-cost sensor network and proposes several efficient and reliable algorithms to address the aforementioned problem.

The primary issue of using low-cost sensors in distributed target tracking is how to balance between communication cost and convergence performance. To overcome this difficulty, we develop a new sample greedy gossip distributed Kalman filter for distributed single-target tracking over a low-cost sensor network. The proposed algorithm leverages the information weighted fusion concept and a new sample greedy gossip process. Instead of finding the optimal path of each node in a greedy manner, the proposed approach utilises a suboptimal communication path by performing greedy selection among randomly selected active sensor nodes. Theoretical convergence analysis and uniform boundedness are also performed to support the proposed algorithm. The main feature of the new algorithm is that it provides fast convergence rate with relatively low communication overload.

In distributed multi-target tracking, each sensor node runs a local multi-target tracking filter for multi-target state and identity estimation. The issue of current multi-target tracking algorithms is that they usually suffer from exponential complexity and require prior knowledge on the environment, e.g., target detection probability and clutter rate. This hinders the application of low-cost sensors in multi-target tracking as these sensors usually have limited computational capability and offline calibration is also not cost-effective. To address these sensor-related practical issues, we propose a polynomial-time joint probabilistic data association filter using stochastic Gibbs sampling technique and incorporate it with a multi-Bernoulli filter to accommodate the unknown environmental parameters. It is theoretically proved that the proposed solution provides a performance-guaranteed approximation and is demonstrated to show promising performance in a dynamic environment.

We then consider the problem of accurate Gaussian mixture approximation for implementing joint probabilistic data association filter when targets are moving closely and propose a novel information-driven approach to tackle this problem. The proposed information-driven joint probabilistic data association algorithm is obtained

from the minimisation of a weighted Kullback-Leibler divergence to approximate the posterior Gaussian mixture probability density function. The resulting approximation approach turns out to show similar structure as the generalised covariance intersection in sensor fusion. Theoretical analysis reveals that the proposed approach with ideal detection probability guarantees boundedness of the error covariance and yields unbiased estimation.

Different from single-target tracking, local multi-target estimations contain data association uncertainty and therefore this issue requires careful adjustment in sensor fusion. Through an equivalent information form, we extend conventional fusion strategies, i.e., consensus on measurement and consensus on information, to multi-target tracking scenarios using joint probabilistic data association filter. This means that data association uncertainty and sensor fusion problems are solved simultaneously. Motivated by the complementary characteristics of these two different fusion approaches, a novel distributed multi-target tracking algorithm using a hybrid fusion strategy, e.g., a mix between consensus on measurement and consensus on information, is proposed. A distributed counting algorithm is incorporated into the tracker to provide the knowledge of the total number of sensor nodes. The new algorithm developed shows advantages in preserving boundedness of local estimates, guaranteeing global convergence to the optimal centralised version and being implemented without requiring global information, compared with other fusion approaches.

To support the implementation of distributed multi-target tracking, we finally investigate the problem of practical implementation of multi-dimensional assignment. To address this problem, we propose two efficient implementation algorithms using Tabu search and Gibbs sampling. As the first step, we formulate the problem of generating the best global hypothesis in multi-dimensional assignment as the problem of finding a maximum weighted independent set of a weighted undirected graph. Then, the meta-heuristic Tabu search with two basic movements is designed to find the global optimal solution of the problem formulated. To improve the computational efficiency, we also develop a sampling based algorithm using Gibbs sampling. The problem formulated for the Tabu search based algorithm is reformulated as a max product problem to enable implementation of Gibbs sampling. The detailed algorithm is then designed and the convergence is also theoretically analysed. The performance of the two algorithms proposed are verified through numerical simulations and compared with that of a mainstream Lagrangian relaxation implementation algorithm.

Acknowledgements

I would like to express my sincere gratitude to my supervisors Prof. Hyo-Sang Shin and Prof. Antonios Tsourdos for their appreciable inspiration, support and patience throughout my research and acting as a mentor to my overall professional development. They provided me excellent opportunities for attending international technical conference and interacting with external research group members. The critical reviews and recommendations that they provided for the improvement of my papers and presentations were very valuable. They were always there for me in every ups and downs of my academic endeavour throughout my PhD life. They were great supervisors and friends and it was a great pleasure to have the opportunity to work under their supervision.

I am really grateful to Prof. Chang-Hun Lee (now at KAIST, South Korea), who has helped my research with insightful and valuable suggestions to improve the quality of my work when he was working at Cranfield University. A special word goes to Dr. Juhyeon Hong, who has spent a lot of time helping me addressing my problems related to the experimental work. She has always been inspiring and supportive when we worked together.

I would like to thank all Autonomous and Intelligent Systems Group members: Sangjun, Inmo, Teng, Hae-In, On, Mingfeng, Wenshan, Minguk, Byoung-Ju, Dmitry, Charles, Yunha, Shuoyuan. Almost every lunch we had an exciting and happy chatting to share our culture, knowledge and other personal opinions. I would also like to thank my friends on campus (the list is long) for making my PhD life a wonderful experience.

Of course, I offer my deepest gratitude to my parents for their endless care, innumerable prayers and sacrifices, and their encouragement which provided me with an immense inspiration.

Lastly, I would like to thank China Scholarship Council for financially supporting my study in UK.

Contents

Contents	v
List of figures	xi
List of tables	xiii
Abbreviations	xv
Notations	xvii
1 Introduction	1
1.1 Background and Motivation	1
1.2 Aim and Objectives	3
1.3 Contributions	5
1.4 Organisation	6
1.5 Publication	7
2 Literature Review	11
2.1 Introduction	11
2.2 Distributed Single Target Tracking	11
2.3 Distributed Multi-Target Tracking	20
2.3.1 Local Multi-Target Tracking	20
2.3.2 Distributed Multi-Target Fusion	28
2.4 Conclusions	30

3	Sample Greedy Gossip Distributed Kalman Filter	31
3.1	Introduction	31
3.2	Sample Greedy Gossip Algorithm	32
3.3	Analysis of Sample Greedy Gossip	35
3.3.1	Asymptotic Convergence Analysis	35
3.3.2	Convergence Rate Analysis	37
3.3.3	Convergence Rate Comparison	44
3.4	Distributed Kalman Filter Design and Analysis	47
3.4.1	Filter Development	47
3.4.2	Boundedness Analysis	50
3.5	Numerical Simulations	53
3.5.1	Simulation Setup	54
3.5.2	Performance Metric	55
3.5.3	Comparison with Different Gossip Algorithms	56
3.5.4	Comparison with Previous Gossip-Based Distributed Estimators	59
3.6	Conclusions	60
4	Practical Implementation of Joint Probabilistic Data Association	62
4.1	Introduction	62
4.2	System Models	64
4.2.1	Multi-Target State and Measurement Model	64
4.2.2	Target Existence Model	64
4.3	Joint Probabilistic Data Association Filter	66
4.3.1	Track Update of JPDA	66
4.3.2	Implementation Issues	69
4.4	Approximated Marginalisation by Stochastic Gibbs Sampling	72
4.4.1	Sampling Joint Association Hypotheses	72
4.4.2	Convergence of Gibbs Sampler	75

4.5	JPDA with Unknown Detection Probability and Clutter Rate	77
4.5.1	Detection Probability and Clutter Rate Estimation	77
4.5.2	MB-JPDA Algorithm	80
4.6	Numerical Simulations	81
4.6.1	Simulation Setup	81
4.6.2	Performance Metric	83
4.6.3	Performance of Gibbs-JPDA	83
4.6.4	Performance of MB-JPDA	88
4.7	Conclusions	90
5	Information-Theoretic Joint Probabilistic Data Association	93
5.1	Introduction	93
5.2	Information-Theoretic JPDA	95
5.2.1	Information-Theoretic Mixture Approximation	95
5.2.2	Implementation of Information-Theoretic JPDA	101
5.3	Performance Analysis	102
5.4	Simulation Studies	106
5.4.1	Simulation Setup	106
5.4.2	Performance Evaluation	108
5.5	Conclusions	108
6	Distributed Joint Probabilistic Data Association Filter with Hybrid Fusion	111
6.1	Introduction	111
6.2	Centralised JPDA Filter: A Benchmark	112
6.2.1	Information-Form JPDA	112
6.3	Distributed JPDA Filter Using Consensus on Measurement and Consensus on Information	116
6.3.1	Average Consensus	116

6.3.2	Distributed JPDA Filter Using Consensus on Measurement . .	117
6.3.3	Distributed JPDA Filter Using Consensus on Information . . .	120
6.4	Distributed JPDA Filter Using Hybrid Fusion Strategy	123
6.4.1	Distributed JPDA Filter Using Hybrid Consensus	123
6.4.2	Algorithm Analysis	127
6.5	Numerical Simulations	129
6.5.1	Simulation Setup	129
6.5.2	Performance Metric	131
6.5.3	Simulation Results	132
6.5.4	Experiments on Real-Life Data	136
6.6	Conclusions	137
7	Practical Implementation of Multi-Dimensional Assignment	139
7.1	Introduction	139
7.2	Preliminaries	140
7.2.1	MDA in Measurement-to-Track Association	141
7.2.2	MDA in Track-to-Track Association	143
7.2.3	Hypothesis Pruning	145
7.3	MWIS Formulation	145
7.4	Solving MDA Problem Using Heuristic Tabu Search	147
7.4.1	Initial Solution Generation	147
7.4.2	Local Search Phase	148
7.5	Solving MDA Problem Using Stochastic Gibbs Sampling	151
7.6	Numerical Simulations	154
7.6.1	Simulation Setup	154
7.6.2	Performance Metric	157
7.6.3	Simulation Results	157
7.7	Conclusions	160

8	Conclusions and Future Works	161
8.1	Conclusions	161
8.2	Future Works	163
	References	164
A	Pseudo Codes of RG-DKF and GG-DKF	177
B	Properties of Finite-State Markov Chain	179
C	Boundedness of Kalman Filter	181
D	Some Approximated Marginalisation Algorithms	183
D.1	ENNJPDA [1]	183
D.2	Modified Cheap JPDA [2]	184
D.3	Suboptimal JPDA [3]	184

List of Figures

1.1	Examples of recent applications of low-cost sensor networks.	2
1.2	Examples of different multi-sensor fusion architectures.	3
1.3	Relationships and links between chapters.	7
2.1	General information flow of distributed MTT.	20
2.2	An example of measurement origin uncertainty	21
2.3	An example of data association.	22
3.1	Examples of two different network topologies.	54
3.2	Monte-Carlo comparison results w.r.t. different number of gossip iterations.	57
3.3	Monte-Carlo comparison results w.r.t. different sensor activation probabilities.	58
3.4	Monte-Carlo comparison results of different gossip DKFs.	60
4.1	Mean average running time of standard JPDA for a typical scenario. .	70
4.2	General information flow of MB-JPDA.	81
4.3	Snapshots of the considered scenario.	82
4.4	Monte-Carlo comparison results w.r.t. different number of targets. . .	84
4.5	Monte-Carlo comparison results w.r.t. different number of clutters. .	85
4.6	Monte-Carlo comparison results w.r.t. different number of Gibbs samples.	86
4.7	Monte-Carlo comparison results w.r.t. different number of burn-in samples.	87

4.8	Monte-Carlo comparison results for a scenario with 10 targets.	88
4.9	Monte-Carlo comparison results for a scenario with 40 targets.	89
4.10	Monte-Carlo comparison results for a scenario with 80 targets.	89
4.11	Monte-Carlo comparison results of mean OSPA distance for different cases.	90
4.12	Monte-Carlo comparison results of mean cardinality estimation for different cases.	91
5.1	An example of closely-moving targets.	94
5.2	Comparison of different Gaussian mixture approximations.	98
5.3	Sample ground truth and measurements.	107
5.4	Mean OSPA distance of 100 Monte-Carlo simulations.	109
6.1	Snapshots of the considered scenario.	129
6.2	Average estimation error w.r.t. number of consensus iterations.	133
6.3	Average estimation error w.r.t. number of sensors.	133
6.4	Average estimation error w.r.t. sensing range.	134
6.5	Average estimation error w.r.t. number of targets.	135
6.6	Average estimation error w.r.t. number of clutters per scan.	135
6.7	Snapshots of EPFL laboratory dataset from four different cameras.	136
6.8	Mean position error averaged over four cameras and four persons.	137
7.1	An example of track hypothesis tree for MDA formulation.	142
7.2	Hypothesis history of the considered scenario.	146
7.3	Undirected graph of the considered example.	146
7.4	Undirected graph of the considered example.	149
7.5	Sample ground truths of the considered scenarios.	155
7.6	OSPA distance with MDA-MHT (green), Tabu-MHT (red) and Gibbs-MHT (blue).	158

List of Tables

2.1	Characteristics of different distributed single target estimators.	19
2.2	Characteristics of different local multi-target estimators.	27
4.1	Total number of joint association hypotheses.	70
4.2	Mean OSPA distance and running time w.r.t. number of targets. . . .	85
4.3	Mean OSPA distance and running time w.r.t. number of clutters. . . .	86
4.4	Mean OSPA distance and running time w.r.t. number of Gibbs samples.	87
4.5	Mean OSPA distance and running time w.r.t. number of burn-in samples.	88
4.6	Comparison results of different approximated marginalisation algo- rithms.	90
4.7	Mean OSPA distance for different cases.	91
5.1	Computational complexity comparison.	101
7.1	Comparison results of Monte-Carlo simulations.	159

Abbreviations

ATHP	approximated track hypothesis probability
AICF	average information consensus filter
CM	consensus on measurement
CMKF	consensus on measurement Kalman filter
CKF	centralised Kalman filter
CI	consensus on information
CI-DDKF	covariance intersection diffusion distributed Kalman filter
CIKF	consensus on information Kalman filter
CPHD	cardinalized probability hypothesis density
CT	coordinate turning
CV	constant velocity
DJPDA	distributed joint probabilistic data association
DDKF	diffusion distributed Kalman filter
DHIF	distributed hybrid information fusion
DKF	distributed Kalman filter
EKF	extended Kalman filter
ENNJPDA	exact nearest neighbourhood joint probabilistic data association
GDKF	gossip distributed Kalman filter
GG	greedy gossip
GICF	generalised interactive Kalman filter
GKCF	generalised Kalman consensus filter
GNN	global nearest neighbourhood
HC	hybrid consensus

ICF	Information consensus filter
JPDA	joint probabilistic data association
JIPDA	joint integrated probabilistic data association
KCF	Kalman consensus filter
KLD	Kullback-Leibler divergence
MAP	maximum a posterior
MDP	max-product decoding
MB-JPDA	multi-Bernoulli joint probabilistic data association
MDA	multi-dimensional assignment
ME	mean error
M-GICF	modified generalised interactive Kalman filter
MHT	multiple hypothesis tracking
MMSE	minimum mean square error
MVF	measurement vector fusion
MWIS	maximum weighted independent set
MTT	multi-target tracking
OSPA	optimal sub-pattern assignment
P-CMCI	parallel consensus on measurement and consensus on information
PHD	probability hypothesis density
PDF	probability density function
PPP	poisson point process
RFS	random finite set
RG	randomised gossip
RMSE	root mean square error
SGG	sample greedy gossip
STT	single target tracking
UKF	unscented Kalman filter

Notations

a, b, c	the lower-case regular type denotes scalar
$\mathbf{a}, \mathbf{b}, \mathbf{c}$	the lower-case bold type denotes vector
$\mathbf{A}, \mathbf{B}, \mathbf{C}$	the upper-case bold type denotes matrix
$\mathcal{A}, \mathcal{B}, \mathcal{C}$	the upper-case calligraphic regular type denotes set
$\mathbf{A}^T$	the transpose of matrix $\mathbf{A}$
$\mathbf{A}^{-1}$	the inverse of matrix $\mathbf{A}$
$\mathbf{I}_n$	the $n \times n$ identity matrix
$\mathbf{0}_n$	the n dimensional column vector of all zero elements
$\mathbf{A} \otimes \mathbf{B}$	the Kronecker product of matrices $\mathbf{A}$ and $\mathbf{B}$
$\ \mathbf{x}\ $	the Euclidean norm vector $\mathbf{x}$
$\text{diag}(\mathbf{x})$	the square diagonal matrix with the elements of vector $\mathbf{x}$ as its main diagonal, and zero elements outside the main diagonal
$ \mathcal{A} $	the cardinality, i.e., the number of elements, of set $\mathcal{A}$
$\Pr(\cdot)$	probability
$p(\cdot)$	probability density function
$\mathbb{E}[\cdot]$	expectation operator
$\mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	real-valued Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$
$\sim$	distributed according to
$\propto$	proportional to
$\int f(\mathbf{x}) d\mathbf{x}$	$\int \int \cdots \int f(x_1, x_2, \cdots, x_n) dx_1 dx_2 \cdots dx_n$
$(\cdot)_{k k-1}$	the prediction of quantity $(\cdot)$ at time instant k
$(\cdot)_{k k}$	the update of quantity $(\cdot)$ at time instant k

Chapter 1

Introduction

1.1 Background and Motivation

There has been an ever-increasing interest in utilising wireless sensor networks for target(s) tracking or estimation in recent decades, driven by its versatility and diverse range of recent applications, including environmental monitoring [4], habitat monitoring [5], airborne target tracking [6], space situation awareness [7], spacecraft navigation [8], etc. The availability of low-cost sensors has enabled the employment of multiple sensor nodes to cooperatively perform large-scale sensing tasks, which are otherwise difficult to accomplish by individually operating these sensing devices. The challenge of operating low-cost sensors, e.g., visual camera, infrared/laser range finder, acoustic sensor, etc, is that they are likely to contain some degree of uncertainties and usually have limited spatial coverage, communication and computation capabilities. Combined with the inherent uncertainties and complexity of the problem, the poor performance issue with these sensors could be significantly exacerbated in target tracking, especially in multi-target tracking (MTT).

It is known that leveraging proper fusion algorithms over the sensor network could synergistically merge the redundant information and effectively complement the deficiencies of each sensor node, hence providing the possibility to improve the tracking and perception performance. Multi-sensor information fusion in wireless sensor networks generally refers to the process of combining sensory data, e.g., range, bearing angle, time of arrival, etc, from several local sensor nodes, such that the resulting perception is in some sense better than when these sensors are used individually for sensing. For example, when operating an autonomous vehicle, see Fig. 1.1 (a) for

example, in an uncertain or unknown dynamic environment, it is necessary to fuse the data from a variety of different sensors, e.g., camera, range finder, so that an accurate estimation of the landmarks and obstacles can be obtained to assist safe and reliable navigation [9]. In situation awareness, the swarm of heterogeneous autonomous robots, capable of communicating with each other, are usually leveraged to cooperatively localise and track targets of interest in an attempt to improve the perception performance, see Fig. 1.1 (b)¹ for example.

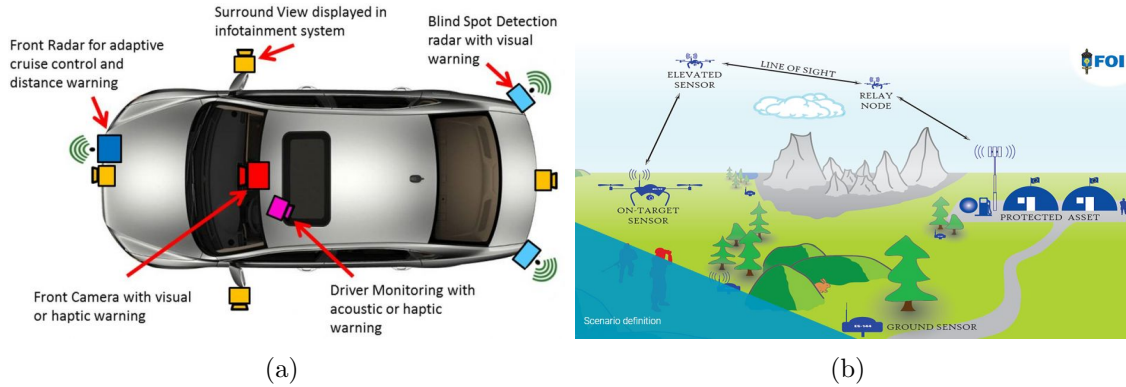


Figure 1.1: Examples of recent applications of low-cost sensor networks in distributed estimation. (a) An autonomous vehicle utilises multi-sensor fusion for navigation. (b) A swarm of heterogeneous autonomous robots cooperatively track targets of interest.

Generally, multi-sensor information fusion can be categorised into three architectures: centralised, decentralised and distributed [10, 11, 12, 13]. Examples of different fusion architectures are shown in Fig. 1.2. In the centralised fusion architecture, all sensors broadcast their local measurements to a fusion centre via single-hop or multi-hop communications. The fusion centre simultaneously processes the measurements provided by all sensors to update the estimate. Unlike centralised fusion, the decentralised architecture utilises several fusion centres, capable of communicating with their neighbours, as backups in data integration, thus showing less vulnerable against system failure. The sensors are allocated to these fusion centres either statically or dynamically depending on the application scenarios [14]. Although multi-sensor fusion through a fusion centre is ideally Bayesian optimal in terms of tracking performance, this architecture normally requires very reliable sensors, which are generally very expensive, and is not scalable. Furthermore, the fusion centre cannot effectively communicate with all sensors for large-scale sensor networks because of physical constraints, e.g., communication delay, limiting com-

¹This application scenario is taken from our previous project *EuroSwarm*. Detailed descriptions of this project can be found at: <https://www.cranfield.ac.uk/research-projects/euroswarm-developing-technology-for-uav>.

munication bandwidth. Each sensor node in the distributed architecture performs fusion using the information only obtained from locally connected neighbours. This could provide enhanced built-in redundancy, which can improve robustness against sensor failure, compared with the other two types of architectures. The distributed fusion architecture can also lower the processing load and communication burden since data is not required to be transmitted to the processing centre and is fused in a distributed way over multiple local nodes. Unlike centralised architecture, the distributed fusion only requires the network to be partially connected and therefore is more flexible to be deployed.

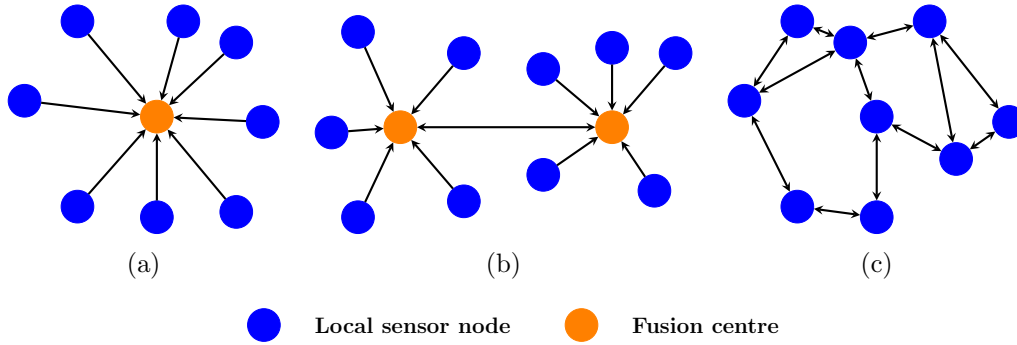


Figure 1.2: Examples of different multi-sensor fusion architectures. The circle denotes the local sensor node and the solid lines refer to the communication between one local sensor and the fusion centre or two sensors. (a) Centralised fusion architecture: all sensor nodes are connected the fusion centre. (b) Decentralised fusion architecture: sensor nodes are allocated to several fusion centres either statically or dynamically. (c) Distributed fusion architecture: sensor nodes only communicate with their neighbours in a peer-to-peer fashion.

Distributed target(s) tracking is usually composed of two stages: (1) local target(s) tracking; and (2) distributed fusion. Each sensor node runs a proper local target(s) tracking algorithm to estimate the state of target(s) from its own noisy measurements. The locally-connected sensor nodes cooperatively communicate with each other by broadcasting their local estimates. Each sensor node then has the possibility of producing higher quality of target(s) estimations by fusing available information from its neighbours through a proper fusion strategy.

1.2 Aim and Objectives

This thesis focuses on distributed target tracking over a low-cost sensor network. As will be discussed in details in Chapter 2, there exist some research challenges

and gaps which need to be addressed. The aim of the thesis is to advance the state-of-the-art distributed target tracking algorithms over a low-cost sensor network by resolving some of the key research challenges and gaps. The main research challenges and gaps to be addressed can be classified into the following four categories:

- Communication issue;
- Data association issue;
- Practical implementation issue;
- Distributed fusion issue.

Responding to the aim to deliver, the following objectives are defined:

- **Reducing the communication overload in distributed fusion:** It is known that iterative communication is usually required to improve the estimation performance in distributed fusion. However, low-cost sensors usually have limited communication ability due to their physical constraints and therefore reducing the communication burden of each local sensor node with limited tracking performance loss is the primary foundation of applying low-cost sensors in distributed estimation.
- **Reducing the time complexity of data association in MTT:** Data association is widely-accepted technique in finding the origins of measurements and local estimates in distributed MTT. However, the Bayesian optimal solutions of typical probabilistic data association problems suffer from exponential complexity and different approximations, depending on the type of data association, are required for practical implementation. For this reason, it is of paramount importance to suggest a unified approach to reduce the time complexity of distributed MTT to enable practical application of low-cost sensors in a cooperative estimation mission.
- **Relaxing the assumption on the full knowledge of environment-dependent parameters in MTT:** The implementation of typical MTT algorithms require the information on target detection probability and clutter rate. However, these two parameters actually depend on the mission scenarios and are not cost-effective to be offline calibrated for low-cost sensors. It is, therefore, meaningful to develop practical MTT algorithms that can accommodate this practical issue.

- **Addressing the issue related to closely-spaced targets in MTT:** For some scenarios where targets move closely, the tracking performance of typical MTT algorithms will degrade due to the nature of approximation made by soft decisions in data association. Therefore, developing efficient MTT algorithms that can mitigate this target-related issue will bring significant benefits for MTT in a complex scenario.
- **Considering data association uncertainty in distributed MTT:** It is known that local MTT estimates usually contain data association uncertainty, which is resulted from the unknown mappings between targets and measurements. This issue becomes more severe for low-cost sensors as these devices are usually subject to low target detection probability and high clutter rate. For this reason, how to properly adjust this uncertainty when fusing local estimates is an important issue for distributed MTT.

1.3 Contributions

This thesis investigates the problem of distributed target tracking over a low-cost sensor network. The key contribution of the thesis is that we address the main research challenges and gaps discussed in Chapter 2 by developing efficient and reliable algorithms. The detailed contributions are summarised as below:

- **Sample greedy gossip distributed Kalman filter (SGG-DKF):** An algorithm for distributed single-target tracking (STT) over a low-cost sensor network is proposed. The promising feature of the proposed SGG-DKF algorithm lies in that it provides great flexibility and balance the communication cost and tracking performance: it provides a relatively fast convergence speed, compared to randomised gossip (RG), with low communication burden, compared to greedy gossip (GG). Another advantage of this new algorithm is that it guarantees local consistency and asymptotic convergence to the Bayesian optimal centralised Kalman filter (CKF).
- **Developing a unified data association approach for distributed MTT:** We propose a unified sampling-based approach for implementing data association in both local MTT phase, i.e., marginalisation in joint probabilistic data association (JPDA), and multi-sensor fusion stage, i.e., multi-dimensional assignment (MDA) in track-to-track association. Specifically, the stochastic

Gibbs sampling is leveraged to generate enough samples of association hypothesis and then the data association problem can be solved by accountable approximations. This new algorithm provides polynomial-time and performance-guaranteed solutions of the original NP-hard data association problem and this enables the employment of low-cost sensors in distributed MTT.

- **Implementation of JPDA without environmental parameters:** We propose a practical implementation algorithm for JPDA without full knowledge of detection probability and clutter rate. The proposed algorithm is derived by integrating a sampling-based JPDA filter and a multi-Bernoulli filter. This new algorithm utilises a closed feedback loop structure based on the property of the posterior of the JPDA estimations and is demonstrated to have comparable performance with ideal JPDA that is assumed to have perfect knowledge of detection probability and clutter rate.
- **Information-theoretic joint probabilistic data association:** A new variant of JPDA algorithm for local MTT is proposed to improve the performance of original JPDA when targets are moving closely. The proposed approach leverages a weighted inverse Kullback-Leibler divergence (KLD) to approximate the posterior Gaussian mixture probability density function (PDF). This provides stable and unbiased estimation that alleviates the multi-modal issue while providing a certain level of robustness against clutters.
- **Distributed joint probabilistic data association with hybrid fusion:** We propose a fully distributed MTT algorithm that automatically considers data association uncertainty in the fusion process. The new algorithm developed shows advantages in preserving boundedness of local estimates, guaranteeing global convergence to the optimal centralised version and being implemented without requiring global information, compared with other fusion approaches.

1.4 Organisation

The rest of thesis is organised as follows. In Chapter 2, we provide a comprehensive review of state-of-the-art solutions for distributed estimation over a low-cost sensor network. In Chapter 3, a distributed STT algorithm, termed as SGG-DKF, is proposed and its convergence performance is theoretically analysed. In Chapter 4,

we introduce the proposed sampling-based JPDA and multi-Bernoulli JPDA (MB-JPDA) for efficient and performance-guaranteed implementation of JPDA under a dynamic environment. In Chapter 5, an information-theoretic JPDA is proposed for local MTT when targets are moving closely. In Chapter 6, we develop a fully distributed algorithm for MTT over a large-scale sensor network, consisting of low-cost sensors. In Chapter 7, the problem of practical implementation of MDA with performance-guaranteed approximation is addressed by both heuristic and stochastic sampling approaches. Chapter 8 offers conclusions reached in the thesis and provides several potential directions for future work. The relationships and links between chapters are summarised in Fig. 1.3.

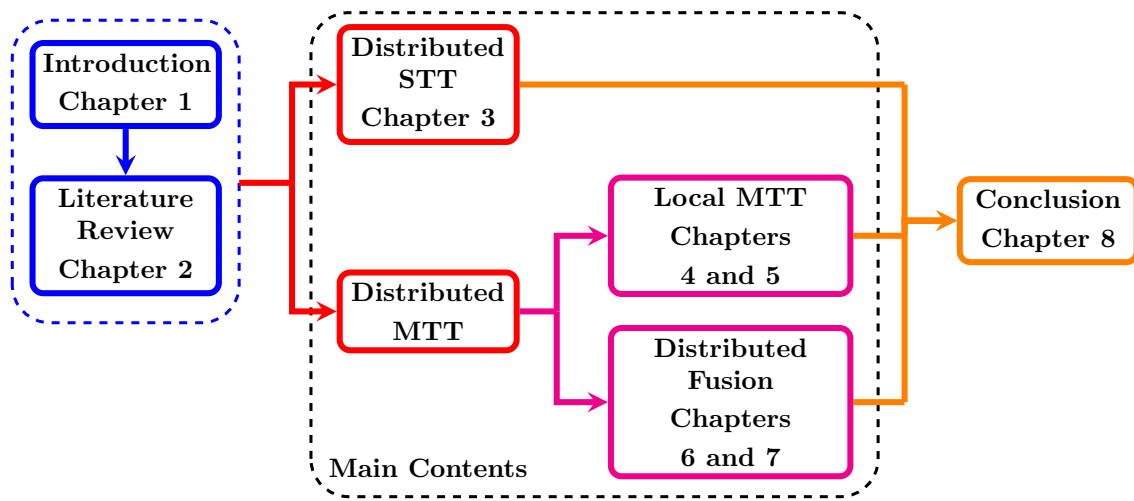


Figure 1.3: Relationships and links between chapters.

1.5 Publication

The following papers are generated during this PhD study.

Published/Accepted

- [1] Shaoming He, Hyo-Sang Shin, Shuoyuan Xu and Antonios Tsourdos. Distributed Estimation Over a Low-Cost Sensor Network: A Review of State-of-the-Art. *Information Fusion*, 54, 21–43, 2020.
- [2] Shaoming He, Chang-Hun Lee, Hyo-Sang Shin, and Antonios Tsourdos. Minimum-Effort Waypoint-Following Guidance. *Journal of Guidance, Control and Dynamics*, 42(7), pp. 1551–1561, 2019.

- [3] Shaoming He, Hyo-Sang Shin, and Antonios Tsourdos. Distributed joint probabilistic data association filter with hybrid fusion strategy. *IEEE Transactions on Instrumentation and Measurement*, 2019, DOI: 10.1109/TIM.2019.2894048.
- [4] Shaoming He, Hyo-Sang Shin, and Antonios Tsourdos. Trajectory Optimization for Target Localization with Bearing-Only Measurement. *IEEE Transactions on Robotics*, 35(3), pp. 653–668, 2019.
- [5] Shaoming He, Hyo-Sang Shin, and Antonios Tsourdos. Trajectory Optimization for Multi-Target Tracking Using Joint Probabilistic Data Association Filter. *Journal of Guidance, Control and Dynamics*, 2019, DOI: 10.2514/1.G004249.
- [6] Shaoming He, Hyo-Sang Shin, Antonios Tsourdos and Chang-Hun Lee. Energy-Optimal Waypoint-Following Considering Autopilot Dynamics. *IEEE Transactions on Aerospace and Electronic Systems*, , 2019, DOI: 10.1109/TAES.2019.2954149.
- [7] Shaoming He, Hyo-Sang Shin, and Antonios Tsourdos. Multi-Sensor Multi-Target Tracking Using Domain Knowledge and Clustering. *IEEE Sensors Journal*, 18(9), pp. 8074–8084, 2018.
- [8] Shaoming He, Hyo-Sang Shin, and Antonios Tsourdos. Track-Oriented Multiple Hypothesis Tracking Based on Tabu Search and Gibbs Sampling. *IEEE Sensors Journal*, 18(1), pp. 328–339, 2018.
- [9] Shaoming He, Hyo-Sang Shin, and Antonios Tsourdos. Joint Probabilistic Data Association Filter with Unknown Detection Probability and Clutter Rate. *Sensors*, 18(1), 269, 2018.
- [10] Hyo-Sang Shin, Dario Turchi, Shaoming He, and Antonios Tsourdos. Behavior Monitoring Using Learning Techniques and Regular-Expressions-Based Pattern Matching. *IEEE Transactions on Intelligent Transportation Systems*, 20(4), pp. 1289–1302, 2018.
- [11] Shaoming He, Hyo-Sang Shin, and Antonios Tsourdos. Computational Guidance Using Sparse Gauss-Hermite Quadrature Differential Dynamic Programming. In *Proceedings of the 21st IFAC Symposium on Automatic Control in Aerospace*, Cranfield, UK, 2019.
- [12] Shaoming He, Hyo-Sang Shin, and Antonios Tsourdos. Constrained Multiple Model Bayesian Filtering for Target Tracking in Cluttered Environment. *IFAC-PapersOnLine*, 50(1), pp. 425–430, 2017.
- [13] Shaoming He, Hyo-Sang Shin, and Antonios Tsourdos. Joint probabilistic data association filter with unknown detection probability and clutter rate. In

Proceedings of the 2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Daegu, Korea, pp. 559–564, 2017.

- [14] Shaoming He, Hyo-Sang Shin, and Antonios Tsourdos. Optimal Active Target Localisation Strategy with Range-Only Measurements. In *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, Prague, Czech Republic, 2019.

Submitted

- [1] Shaoming He, Hyo-Sang Shin, and Antonios Tsourdos. Information-Driven Joint Probabilistic Data Association Filter. *IEEE Transactions on Automatic Control*, Revision Submitted.
- [2] Shaoming He, Hyo-Sang Shin, and Antonios Tsourdos. Distributed Multiple Model Joint Probabilistic Data Association with Gibbs Sampling-Aided Implementation. *Information Fusion*, Revision Submitted.
- [3] Hyo-Sang Shin, Shaoming He and Antonios Tsourdos. Sample Greedy Gossip for Distributed Network-Wide Average Computation. *IEEE Transactions on Information Theory*, Submitted.

Chapter 2

Literature Review

2.1 Introduction

The problem of distributed estimation over a sensor network has been approached by numerous contributions proposed in the past years. The state-of-the-arts are broad and rich, but quite fragmented. Each approach has its own performance advantages and disadvantages depending on the application scenarios. This chapter is an endeavour to review the state-of-the-art solutions of distributed estimation over a low-cost sensor network. For better readability, the review presented in this chapter is split into two parts: (1) distributed STT; and (2) distributed MTT.

2.2 Distributed Single Target Tracking

Distributed STT aims to find the trajectory of a target of interest by using a partially connected sensor network, i.e., a local sensor node can only receive the information from its connected nodes. In STT, it is generally assumed that each sensor node has a unity probability of target detection with zero false alarm rate to simplify the design and analysis. A general distributed STT algorithm usually consists of two components: (1) local Kalman filter or its related algorithms, e.g., extended Kalman filter (EKF), unscented Kalman filter (UKF); and (2) distributed fusion strategy. As local Kalman filter is a well-known and widely-used technique in estimating the current state of a target, this subsection only focuses on the review of distributed fusion strategy.

An optimal fusion strategy and benchmark for performance evaluation of distributed state estimation algorithms is the CKF, which processes all sensors' measurements simultaneously through a fusion centre. It is known that the information form of Kalman filter is a suitable formula to address multi-sensor data fusion problem in a distributed manner. This information-form variant is functionally identical to the original Kalman filter, but has computational advantages for high-dimensional data. Based on the property of estimators with information form, incorporating additional information from other sensors could be achieved by summation of the corresponding information terms. This implies that the update procedure of the Bayesian optimal CKF for a network with N sensors can be formulated as [12]

$$\begin{aligned}\mathbf{P}_{k|k}^{-1}\mathbf{x}_{k|k} &= \mathbf{P}_{k|k-1}^{-1}\mathbf{x}_{k|k-1} + \sum_{i=1}^N \mathbf{H}_{k,i}^T \mathbf{R}_{k,i}^{-1} \mathbf{z}_{k,i} \\ \mathbf{P}_{k|k}^{-1} &= \mathbf{P}_{k|k-1}^{-1} + \sum_{i=1}^N \mathbf{H}_{k,i}^T \mathbf{R}_{k,i}^{-1} \mathbf{H}_{k,i}\end{aligned}\tag{2.1}$$

where $\mathbf{x}_{k|k-1}$ (respectively, $\mathbf{x}_{k|k}$) denotes the prediction (respectively, update) of target state at time instant k and $\mathbf{P}_{k|k-1}$ (respectively, $\mathbf{P}_{k|k}$) is the corresponding error dynamics. The vector $\mathbf{z}_{k,i}$ represents the measurement obtained from the i th sensor. The matrices $\mathbf{H}_{k,i}$ and $\mathbf{R}_{k,i}$, respectively, stand for the measurement matrix and the covariance of measurement noise of the i th sensor.

It is clear that centralised estimation (2.1) requires full information of all sensors. Considering the fact that each sensor usually can only communicate with its neighbours due to communication limit, CKF is generally not applicable in practical low-cost sensor networks. However, the Bayesian optimal centralised solution will be utilised as a benchmark for the performance comparison and evaluation of the distributed STT fusion algorithms discussed in the following subsections.

As there is no processing centre in the distributed fusion architecture, the fundamental problem naturally arises: how to effectively perform information fusion using only neighbours' information? Depending on how the local sensor nodes communicate with their neighbours, three representative distributed fusion strategies have been proposed in the existing literature: consensus protocol, gossip process and diffusion strategy. More specifically, consensus-based fusion rule utilises the control-theoretic consensus algorithm to achieve agreement on local estimates among the sensor nodes; gossip process also achieves consensus on local information, but in a randomised or deterministic way; diffusion strategy aims to provide a convex com-

combination of local information to improve the estimation accuracy. Based on these facts, the reviews and discussions of distributed STT estimation algorithms will be carried out by considering these three different fusion strategies.

Consensus-Based Algorithms

With the development of network theory, the control-theoretic consensus algorithm [15, 16, 17, 18] was found to be a powerful tool in designing distributed estimation filters that guarantee global convergence. This can be attributed to the fact that this algorithm enables performing network-wide computation tasks, such as averaging of quantities and functions. Olfati-Saber [19] suggested a distributed algorithm, termed as Kalman consensus filter (KCF), by performing average consensus on local estimates. This work seems to be the pioneer work in the domain of globally distributed estimation over a sensor network. The stability and performance bounds of KCF were theoretically analysed later in [20]. The fused estimate at the l th consensus iteration step provided by the i th sensor is given as

$$\mathbf{x}_{k|i,i}^{(l)} = \sum_{j \in \mathcal{N}_i} \pi_{k,ij} \mathbf{x}_{k|i,j}^{(l-1)} \quad (2.2)$$

where $\mathcal{N}_i$ denotes the connected local neighbours of sensor i ; the consensus gain $\pi_{k,ij} > 0$ is normally chosen based on the degree of the network graph [21]. Maximum-degree weights and metropolis weights are two widely-used suboptimal consensus gains to achieve average consensus [22]. For simplicity, we will use the operator $\Psi(\cdot)$, instead of the weighted summation form (2.2) as the average consensus protocol in the following discussions.

The limitation of KCF lies in that it weights all local neighbours' prior states equally and thus the performance degrades drastically when some sensors cannot detect the target due to limited field-of-view. The limited sensing range, together with sparse communication network topologies, will have a profound effect on the transient behaviour of KCF and even result in divergent estimation. To mitigate this issue, the authors of [23] proposed a generalised KCF (GKCF) via weighting neighbours' prior states by their corresponding prior covariance matrices. Although GKCF outperforms KCF in terms of estimation accuracy, it cannot guarantee global optimality, i.e., its accuracy does not converge to that of the centralised filter (2.1) even with infinite number of consensus iterations. The reason is that GKCF never utilises the useful local posterior covariance information in sensor fusion. Mainly due to this

fact, both KCF and GKCF are suboptimal.

In order to guarantee global convergence to the centralised estimation, Olfati-Saber [24] utilised the concept of consensus on measurement (CM) in Kalman filter (CMKF). The idea of CMKF is to compute the information terms $\sum_{i=1}^N \mathbf{H}_{k,i}^T \mathbf{R}_{k,i}^{-1} \mathbf{z}_{k,i}$ and $\sum_{i=1}^N \mathbf{H}_{k,i}^T \mathbf{R}_{k,i}^{-1} \mathbf{H}_{k,i}$ in a distributed manner through average consensus to match with the CKF. By exchanging the local information vector $\mathbf{H}_{k,i}^T \mathbf{R}_{k,i}^{-1} \mathbf{z}_{k,i}$ and information matrix $\mathbf{H}_{k,i}^T \mathbf{R}_{k,i}^{-1} \mathbf{H}_{k,i}$, the update rule of CMKF is given by

$$\begin{aligned} \mathbf{P}_{k|k,i}^{-1} \mathbf{x}_{k|k,i} &= \mathbf{P}_{k|k-1,i}^{-1} \mathbf{x}_{k|k-1,i} + N \Psi \left(\mathbf{H}_{k,j}^T \mathbf{R}_{k,j}^{-1} \mathbf{z}_{k,j} \right) \\ \mathbf{P}_{k|k,i}^{-1} &= \mathbf{P}_{k|k-1,i}^{-1} + N \Psi \left(\mathbf{H}_{k,j}^T \mathbf{R}_{k,j}^{-1} \mathbf{H}_{k,j} \right), \quad j \in \mathcal{N}_i \end{aligned} \quad (2.3)$$

The advantage of CMKF is that it is asymptotically optimal at each time instant provided that the priors are converged. However, since only finite number of consensus iterations is tractable in practice, convergence will not be fully achieved. Therefore, all local estimates are auto-correlated during the fusion phase and thus CMKF suffers from the well-known auto-correlation problem. As CMKF never exploits the prior information in fusion, it constrains the posterior estimates as the prior estimates if the sensor and its neighbours cannot detect the target within allowable number of consensus iterations due to limited field-of-view. Also note from Eq. (2.3) that CMKF weights the local prior estimate using its own prior covariance matrix, which means it cannot preserve the consistency of local estimates since the observability condition of local sensor can only be ensured with enough number of iterations. Apparently, this issue becomes more severe for sparse networks. Therefore, the performance of CMKF with small number of consensus iterations will degrade significantly. To solve the associated problems of CMKF, the authors of [25, 26] presented a distributed consensus on information (CI) Kalman filter (CIKF) based on the concept of generalised CI [27] as

$$\begin{aligned} \mathbf{P}_{k|k,i}^{-1} \mathbf{x}_{k|k,i} &= \Psi \left(\mathbf{P}_{k|k-1,j}^{-1} \mathbf{x}_{k|k-1,j} + \mathbf{H}_{k,j}^T \mathbf{R}_{k,j}^{-1} \mathbf{z}_{k,j} \right) \\ \mathbf{P}_{k|k,i}^{-1} &= \Psi \left(\mathbf{P}_{k|k-1,j}^{-1} + \mathbf{H}_{k,j}^T \mathbf{R}_{k,j}^{-1} \mathbf{H}_{k,j} \right), \quad j \in \mathcal{N}_i \end{aligned} \quad (2.4)$$

The CIKF was proven to generate unbiased local estimates and to be equivalent to perform consensus on the Kullback-Leibler average of local PDFs in [28]. Due to the nature of the CI fusion rule, CI-based distributed estimators ensure the consistency of local estimate and hence generally shows better performance than CMKF when the number of consensus iterations is limited. But, unfortunately, CIKF is not globally optimal as it underweights the information related terms $\sum_{i=1}^N \mathbf{H}_{k,i}^T \mathbf{R}_{k,i}^{-1} \mathbf{z}_{k,i}$

and $\sum_{i=1}^N \mathbf{H}_{k,i}^T \mathbf{R}_{k,i}^{-1} \mathbf{H}_{k,i}$. This will generate the so-called local redundancy issue. Although Ellipsoid CI [29] and inverse CI [30] have great potentials to reduce the conservativeness of CI, these two fusion rules currently can only be implemented in a sequential way for multiple sensors.

Comparing CMKF and CIKF, we can observe that these two algorithms have complementary properties: CMKF is asymptotically optimal but its performance degrades significantly when the number of consensus iterations is small; CIKF is beneficial for ensuring the consistency of fused estimates but cannot recover the performance of centralised estimator. Motivated by these observations, the authors of [31, 32] proposed a new distributed estimation algorithm, termed as information consensus filter (ICF). The fusion rule of this filter is given by

$$\begin{aligned} \mathbf{P}_{k|k,i}^{-1} \mathbf{x}_{k|k,i} &= N \Psi \left(\frac{1}{N} \mathbf{P}_{k|k-1,j}^{-1} \mathbf{x}_{k|k-1,j} + \mathbf{H}_{k,j}^T \mathbf{R}_{k,j}^{-1} \mathbf{z}_{k,j} \right) \\ \mathbf{P}_{k|k,i}^{-1} &= N \Psi \left(\frac{1}{N} \mathbf{P}_{k|k-1,j}^{-1} + \mathbf{H}_{k,j}^T \mathbf{R}_{k,j}^{-1} \mathbf{H}_{k,j} \right), \quad j \in \mathcal{N}_i \end{aligned} \quad (2.5)$$

The ICF algorithm is able to converge asymptotically to the centralised filter (2.1). However, it requires some global information, e.g., degree of the sensor network and network size, in implementation. To partially relax this assumption, Yao et al. [33] suggested an average ICF (AICF), which, however, essentially coincides with the original CIKF [25, 26]. A generalised consensus fusion framework that encompasses CMKF, CIKF and ICF was presented in [34] via a parallel CM and CI (P-CMCI) as

$$\begin{aligned} \mathbf{P}_{k|k,i}^{-1} \mathbf{x}_{k|k,i} &= \Psi \left(\mathbf{P}_{k|k-1,j}^{-1} \mathbf{x}_{k|k-1,j} \right) + \omega_{k,i} \Psi \left(\mathbf{H}_{k,j}^T \mathbf{R}_{k,j}^{-1} \mathbf{z}_{k,j} \right) \\ \mathbf{P}_{k|k,i}^{-1} &= \Psi \left(\mathbf{P}_{k|k-1,j}^{-1} \right) + \omega_{k,i} \Psi \left(\mathbf{H}_{k,j}^T \mathbf{R}_{k,j}^{-1} \mathbf{H}_{k,j} \right), \quad j \in \mathcal{N}_i \end{aligned} \quad (2.6)$$

where the normalisation factor $\omega_{k,i} > 0$ is a design parameter to tune the filter performance. Similar to ICF, P-CMCI handles the redundancy of the priors and thus has possibility of global convergence to the centralised solution by setting the normalisation factor as the network size, i.e., $\omega_{k,i} = N$, but requires double communication cost due to the inherent parallel consensus process. The main promising feature of P-CMCI filter is that it has been proved to be able to guarantee stability for any choice of the normalisation weights $\omega_{k,i}$ and for any number of consensus steps.

Gossip-Based Algorithms

Instead of average consensus, Ma et al. [35] developed a gossip DKF (GDKF) by performing RG process to local state estimate. At every round of gossip iteration, each sensor using GDKF randomly selects a locally-connected neighbour node and performs averaging on these two local state estimations. The main positive feature of GDKF is that it has relatively low communication burden since each sensor only needs to communicate with one connected sensor node during one gossip iteration. Compared to GDKF, a sensor node running KCF receives information from all its locally-connected neighbours and hence generates better estimation performance in terms of accuracy at the price of high communication cost. Notice that both KCF and GKCF only utilise local state estimates in the fusion process. This means that GDKF also cannot recover the performance of centralised estimation even with infinite number of gossip iterations. Another benefit of utilising gossip process in distributed estimation is that gossip-based algorithms are applicable to asynchronous fusion. However, gossip-based estimators under asynchronous condition show much slower convergence speed than the synchronous mode [36].

Another gossip-based DKF, termed as a gossip interactive Kalman filter (GIKF), was developed in [37]. The fundamental difference between GIKF and other DKFs is that GIKF runs the consensus and observation updates at the same time scale. At a random time instant, a local sensor node i randomly selects a neighbour $\bar{i} \in \mathcal{N}_i$ to swap their prior estimates and the corresponding error covariances for measurement update as

$$\begin{aligned}\mathbf{x}_{k|k,i} &= \mathbf{x}_{k|k-1,\bar{i}} + \mathbf{K}_{k,i} (\mathbf{z}_{k,i} - \mathbf{H}_{k,i} \mathbf{x}_{k|k-1,\bar{i}}) \\ \mathbf{P}_{k|k,i} &= \mathbf{P}_{k|k-1,\bar{i}} - \mathbf{K}_{k,i} \mathbf{H}_{k,i} \mathbf{P}_{k|k-1,\bar{i}}\end{aligned}\tag{2.7}$$

which guarantees probabilistically global convergence. Li et al. [38] developed a modified GIKF (M-GIKF) to improve the performance of GIKF by adding one additional observation mixing step.

Different from [37, 38], Qin et al. [39] developed a RG-based measurement vector fusion (RG-MVF) algorithm that leverages the RG process in information fusion by replacing the average consensus in CMKF with the RG algorithm. Theoretical analysis reveals that the utilisation of randomised protocols helps to avoid the need of cumbersome communication, thus reducing the need of time to perform sensor fusion. Although the randomised gossip process is proved to guarantee average agreement among all sensor nodes with infinite number of iterations [36], the perfor-

mance of RG-MVF degrades drastically due to its slow convergence, compared to the original CMKF. To address this problem, a deterministic communication strategy using GG was suggested in [40] to develop a new DKF, termed as GG-MVF, that can improve the convergence rate of the gossip process. The idea behind GG-MVF is that it finds the local sensor node for communication that provides the largest information gain via a greedy manner. However, GG-MVF requires each sensor node to communicate with all its connected neighbours to find an optimal path and therefore exhibits higher communication burden than RG-MVF.

Diffusion-Based Algorithms

Unlike consensus and gossip based DKFs, the diffusion strategy does not require the information on the network size and has low communication burden by using single communication step [41, 42]. The authors of [42] first suggested a DKF for a sensor network using diffusion strategy. Unlike KCF and GKCF, the diffusion DKF (DDKF) utilised a single-step convex combination of the estimates of local neighbours as

$$\mathbf{x}_{k|k,i} = \sum_{j \in \mathcal{N}_i} c_{k,ij} \mathbf{x}_{k|k,j} \quad (2.8)$$

Comparing Eqs. (2.2) and (2.8), it can be noted that diffusion is a special case of consensus with one iteration only. It follows from Eq. (2.8) that the fused estimate at every sensor node provided by the diffusion strategy is a linear combination of the estimates available from the connected neighbours. This observation reveals that the scalar weights $c_{k,ij}$ pose significant impact on the fusion performance.

Since the original DDKF only utilised local estimates in information fusion, it has similar drawbacks as KCF. To address this problem, the authors of [43, 44] leveraged covariance intersection to improve the local estimation performance before the diffusion step and therefore the resulting algorithm is termed as CI-DDKF. Another improvement over [44] was reported in [45], where the authors developed a distributed hybrid information fusion (DHIF) filter. This algorithm is composed of two main steps: the first step utilises the batch covariance intersection to fuse local priors, in a similar way as [44], and the second step applies optimal fusion (2.1) to local sensors to update the state estimates. Note that the DHIF can be considered as a special version of P-CMCI [34] with single consensus iteration. The major drawback of diffusion-based algorithms is that the property of asymptotic convergence to the centralised solution is lost.

Summary of Existing Estimation Algorithms

The main characteristics of the aforementioned distributed STT estimators are summarised in Table 2.1. The advantages and disadvantages of the discussed distributed estimation algorithms are compared with respect to different metrics:

- Global optimality¹: the algorithm is able to converge to the Bayesian-optimal centralised solution asymptotically;
- Local consistency: the fused estimate can preserve local consistency, i.e., the actual local covariance is always bounded by the fused covariance;
- Fully distributed: the fusion algorithm requires no global information, e.g., network size, node number;
- Communication burden: the number of communication rounds during the fusion process.

As CMKF [24] and its related variants [39, 40] cannot preserve the local consistency, these algorithms are not preferred if the communication capacity is limited as the performance of these algorithms degrades drastically with small number of consensus or gossip iterations. Since ICF-based approaches [32, 33, 34] are proved to be able to guarantee stability for any number of consensus steps, these algorithms are applicable for generic scenarios with limited communication resource. Another alternative choice to reduce the communication burden is the one-iteration-only diffusion-based methods [42, 44, 43], which can be viewed as special cases of the consensus-based algorithms. Although diffusion-based approaches have low communication cost, consensus or gossip based DKFs usually provide better performance in terms of tracking accuracy, if multiple rounds of communications are allowed in the applications.

From the previous discussions, we can conclude that there is a significant conflict between the fusion performance and communication requirement, as a higher estimation accuracy normally requires more communication resources, either more communication iterations or higher communication bandwidth which are usually limited for low-cost sensors. For this reason, one fundamental question of distributed estimation over a low-cost sensor network is how to well balance between communication overload and tracking performance.

¹Note that when we discuss about the possibility of *global convergence* or *global optimality* for a specific distributed estimation algorithm, we assume that the sensor network is connected (undirected graph) or strongly connected (directed graph). However, this does not mean this algorithm requires the network to be connected or strongly connected in implementation.

Table 2.1: Characteristics of different distributed single target estimators.

Fusion strategy	Algorithm	Global optimality	Local consistency	Fully distributed	Communication burden
Average consensus	KCF [19, 20, 46]	No	No	No	High
	GKCF [23]	No	Yes	No	High
	CMKF [24, 47, 48]	Yes	No	No	High
	CIKF [25, 26, 28]	No	Yes	No	High
	ICF [31, 32]	Yes	Yes	No	High
	AICF [33]	No	Yes	No	High
	P-CMCI [34]	No	Yes	Yes	High
Gossip process	GDKF [35]	No	No	Yes	Medium
	GIKF [37]	No	Yes	Yes	Medium
	M-GIKF [38]	No	Yes	Yes	Medium
	RG-CMKF [39]	Yes	No	Yes	Medium
	GG-CMKF [40]	Yes	No	No	High
Diffusion algorithm	DDKF [42]	No	No	Yes	Low
	CI-DDKF [44, 43]	No	Yes	Yes	Low
	DHIF [45]	No	Yes	Yes	Low

2.3 Distributed Multi-Target Tracking

Unlike STT, the primary objective of MTT is to simultaneously estimate target trajectories and the number of targets. Generally, a distributed MTT algorithm contains two main stages: the local multi-target tracker for each sensor node to obtain local estimates and the distributed information fusion algorithm for locally connected sensors to fuse local estimates. A general information flow of distributed MTT is shown in Fig. 2.1. Compared to distributed STT, distributed MTT introduces extra difficulties in both stages:

- Measurement origin uncertainty in local filtering;
- Data association and track origin uncertainty in distributed fusion.

This subsection will present a review of state-of-the-art distributed MTT algorithms that address these challenge issues and discuss their corresponding limitations.

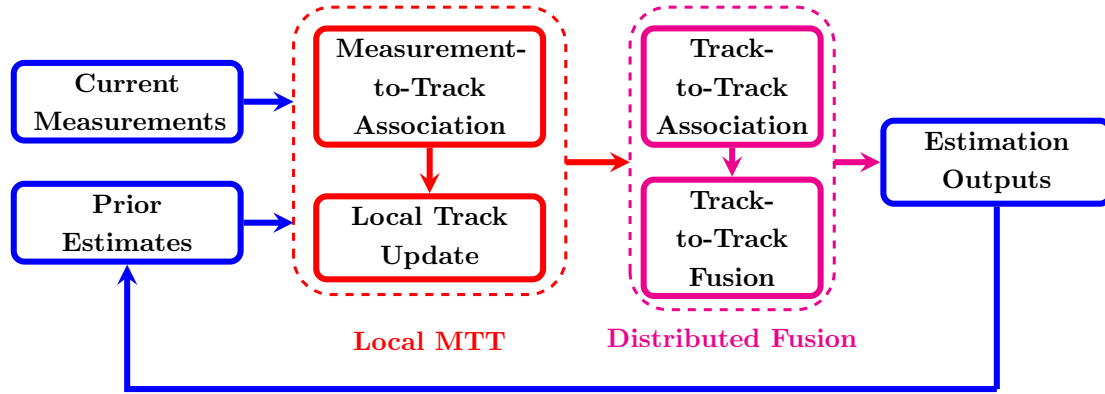


Figure 2.1: General information flow of distributed MTT.

2.3.1 Local Multi-Target Tracking

In a general MTT mission, the relationship between targets and measurements is unknown and the number of targets is also a random variable. Each target can appear and disappear at any place and any time. Also, it is known that low-cost sensors are generally subject to certain degree of uncertainties, meaning that the data quality from such sensing hardware is mainly characterised by reduced accuracy and reliability. For this reason, the utilisation of low-cost sensors often encounters with several challenging issues, including miss detection and false alarms or clutters. This means that the usual assumption on the unity probability of target detection and zero false alarm rate in STT is not valid. This facts makes the

problem of measurement origin more complicated and challenging. Fig. 2.2 gives a general description of measurement origin uncertainty. In this example, there are two confirmed tracks denoted by coloured diamonds and rectangles. The black cross out symbols refer to the observations get from one sensor at the current time instant. Without the knowledge of the relationship between targets and measurements, it is impossible to carry out measurement update using typical filtering techniques.

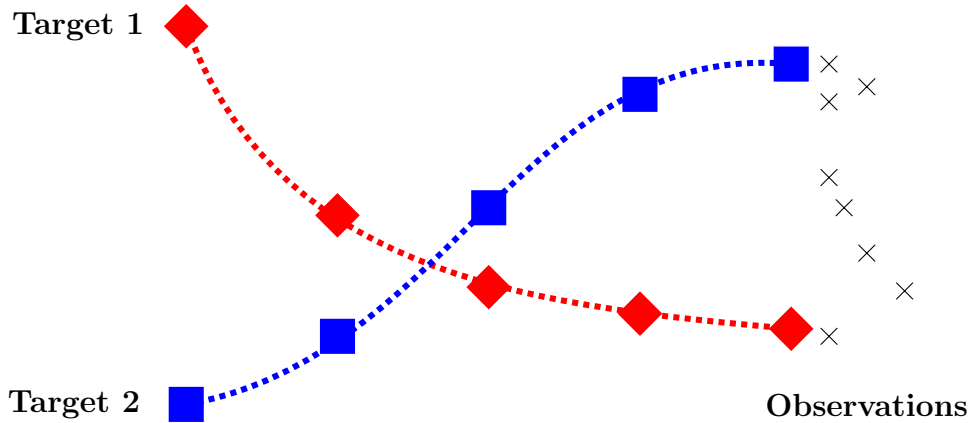


Figure 2.2: An example of measurement origin uncertainty. The coloured diamonds and rectangles refer to two confirmed tracks. The black cross out represents the observation get from one sensor at current time instant.

Data association is a widely-accepted and plausible solution to resolve the problem of measurement origin uncertainty. This technique discerns target-generated measurements from clutters and finds the mappings between targets and measurements, and therefore is the key of local MTT. Depending on the types of decisions, data association can be generally categorised into two main classes: hard decision and soft decision. Hard decision utilises only one specific measurement for updating each track while soft decision considers all possible measurements for each target in track update. The representatives of hard decision are global nearest neighbour (GNN) filter and multiple hypothesis tracking (MHT). As for the soft decision, JPDA is one of the most popular approach for data association.

Global Nearest Neighbour Filter

The most fundamental and conventional data association for MTT is the GNN filter [49, 50]. GNN makes a hard decision and aims to find the observation nearest to each target in terms of Mahalanobis distance, among all observations in the target

validation region². In the GNN filter, all possible associations within the gates are taken into account and GNN finds the best joint association hypothesis that provides the highest probability. Consider an example shown in Fig. 2.3, where the coloured circles represent the predicted measurements based on the confirmed targets and the black ‘ $\times$ ’ symbols stand for measurements. The coloured ellipses denote the validation regions of the corresponding targets. According to the principle of GNN, measurements $\mathbf{z}_k^1, \mathbf{z}_k^2$ will be assigned to target 1 and 2, respectively.

In practical application, the 2D assignment problem formulated by GNN can be efficiently solved by the well-known Hungarian or auction algorithm [50]. To confirm a new track and delete an existing track, GNN leverages a simple heuristic M/N logic³. The GNN filter is computationally efficient and is easy for real implementation. Nonetheless, as this approach prunes all other feasible joint association hypotheses from the association, its performance degrades drastically for nontrivial scenarios, i.e., low the target detection probability or high false alarm rate. For this reason, it is, therefore, more desirable to conduct data association using a probabilistic or Bayesian decision process.

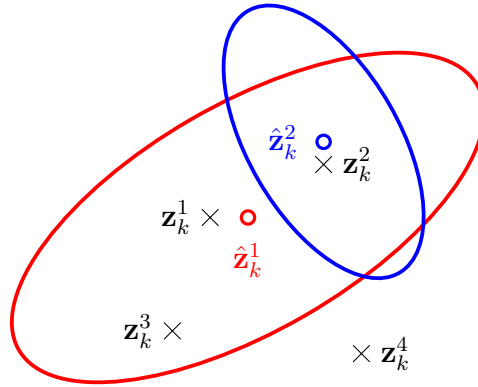


Figure 2.3: An example of data association. The coloured circles represent the predicted measurements based on the confirmed targets and the black ‘ $\times$ ’ symbols stand for measurements. The coloured ellipses denote the validation regions of the corresponding targets.

²Gating is a practical technique to remove the most unlikely measurements to form a target validation region for each track. Only those measurements that are within the validation region are considered for the update of confirmed tracks. Gating usually utilises the χ^2 test and a typical ellipsoidal gate is defined as $(\mathbf{z}_k^j - \hat{\mathbf{z}}_{k|k-1}^i)^T (\mathbf{S}_k^i)^{-1} (\mathbf{z}_k^j - \hat{\mathbf{z}}_{k|k-1}^i) \leq \gamma$ [50], where $\hat{\mathbf{z}}_{k|k-1}^i$ denotes the predicted measurement of the i th target and $\mathbf{S}_k^i$ is the covariance of the innovation term $\mathbf{z}_k^j - \hat{\mathbf{z}}_{k|k-1}^i$. γ is the maximum Mahalanobis distance that controls the volume of the gate.

³During the last N updates, at least M measurements must have been associated with the tentative track.

Joint Probabilistic Data Association

The most well-known one-scan probabilistic data association approach for MTT is the JPDA filter [51]. Unlike GNN, JPDA utilises a Bayesian inference approach to make a soft decision in data association and allows for the possibility that each measurement can come from different target candidates. For example, in the scenario shown in Fig. 2.3, measurements $\mathbf{z}_k^1$, $\mathbf{z}_k^2$ and $\mathbf{z}_k^3$ have possibility to originate from target 1. Based on the principle of JPDA, it will find the marginal association probability of each possible target-measurement pair, i.e., $\mathbf{z}_k^1 \rightarrow \hat{\mathbf{z}}_k^1$, $\mathbf{z}_k^2 \rightarrow \hat{\mathbf{z}}_k^1$ and $\mathbf{z}_k^3 \rightarrow \hat{\mathbf{z}}_k^1$, and utilise the marginal association probabilities as the weights to combine these three different individual updates.

The JPDA filter is also mathematically proved to be a Bayesian optimal estimator in the sense of minimum mean square error (MMSE) performance [52]. Since JPDA forms tracks out of the marginal measurement-to-track association distributions, it is known as a track-oriented approach. The original JPDA paradigm assumes the number of targets is known *a priori* for data association and also leverages the heuristic M/N logic for track management. More recently, target existence models are incorporated in JPDA for Bayesian inference of target confirmation and deletion, as in joint integrated probabilistic data association (JIPDA) [53, 54, 55].

Even though JPDA is demonstrated to be robust against undesirable environmental disturbances, implementing JPDA requires to enumerate all feasible joint association hypotheses to find the marginal association probability. Note that the marginal probabilities are used as weights to update each track from its possible measurement candidates. It is known that computing the marginal probabilities itself is #P-complete, and the time complexity increases exponentially with the increase of the number of targets. This precludes the application of JPDA in low-cost sensors as these devices are generally battery-powered with limited computational ability. To reduce the computational cost of the JPDA, many *ad hoc* formulations and approximations of JPDA, such as cheap JPDA [1], exact nearest neighbourhood JPDA (ENNJPDA) [1], modified cheap JPDA [2], suboptimal JPDA [3] and m -best JPDA [56] were developed in the past few years. The issue with these approximations is that the performance is not guaranteed theoretically, i.e., most of them provide reasonable computational load reduction by sacrificing the tracking performance.

Due to the nature of the soft decision, the posterior distribution provided by JPDA is a Gaussian mixture PDF with each Gaussian term representing one possible measurement-to-target association. Since the propagation of Gaussian mixture is

computationally intractable in the implementation, JPDA approximates the posterior Gaussian mixture by a single Gaussian using simple moment-preserving approach. Clearly, this simple Gaussian mixture reduction method can be utilised to approximate the posterior as long as the targets are well-spaced. If targets are moving closely or in a highly cluttered environment, the posterior Gaussian mixture will be highly bimodal (or multi-modal) and therefore the single Gaussian approximation made by JPDA is not accurate enough to represent the real posterior PDF [57]. This fact means that the JPDA filter tend to ignore the interactions between those closely-spaced targets and the estimation performance will degrade. Instead of considering all possible feasible measurements, the ENNJPDA [1] only picks up the one with the highest marginal association probability for track update. This drastic pruning strategy shows remarkable performance when targets are moving closely, but results in significant information loss as only one measurement is allocated to a track. This means that the ENNJPDA is sensitive to the clutter rate and therefore is not preferred for low-cost sensors. An adjusted version of the JPDA filter, termed as the Set JPDA filter [58], is proved to be an effective way to preserve the multi-modality of the posterior Gaussian mixture PDF. However, this algorithm assumes the number of targets is known a priori and cannot preserve the target identity. By introducing a novel measurement-driven JPDA [59], the multi-modality issue can be directly avoided in track update. The measurement-driven JPDA, however, cannot preserve the track identity and is shown to be sensitive to the measurement noise. Using Bayesian variational approach, the authors in [60] proposed another JPDA to approximate the posterior PDF, but this approach requires an iterative optimisation loop for the implementation and therefore is computationally expensive. From extensive numerical studies, we also found that the performance of this variational JPDA degrades drastically when the measurement is nonlinear with respect to the system state.

Multiple Hypothesis Tracking

Recent dramatic advance in computational capabilities has led to another widely-used data association approach, namely MHT. Unlike GNN and JPDA, MHT utilises several consecutive scans of measurements to solve the data association problem by a delayed logic. Once enough information is available, MHT makes a hard decision using probabilistic inference and this approach is known as a Bayesian optimal maximum a posterior (MAP) estimator. Another benefit of MHT is that it exploits track quality and leverages Bayesian inference in track confirmation and deletion.

From this point of view, MHT is considered to have a fundamental advantage over classical heuristic track maintenance approaches. Another benefit of using multiple scans of measurements in data association is that it can partially solve the ID switch problem when targets' trajectories cross with each other.

Generally, MHT can be categorised into two main types: (1) hypothesis-oriented; and (2) track-oriented. The hypothesis-oriented MHT, originally proposed by Reid [61], forms and expands a large number of hypotheses scan to scan for data association. On the other hand, the track-oriented MHT [62] maintains track trees for incompatible tracks and discards all the unreliable tree branches (track hypotheses) formed on last scans based on the best global hypothesis. In both categories, how to generate the best global hypotheses is a key part for the implementation. The naive brute force enumeration of all hypotheses is impractical for the real-time application as the number of hypotheses exponentially increases. This is the major drawback of MHT for practical applications. In case of the hypothesis-oriented MHT, Cox [63] proposed an efficient method by considering only m -best hypotheses using Murty's ranking algorithm [64]. As to track-oriented MHT, MDA formulation [65, 66, 67] is the mainstream for the implementation. The MDA problem with scan depth of more than two is known to be NP-hard. Solutions of this problem are generally obtained by an approximate algorithm such as Lagrangian relaxation approach [65], m -best assignment [68], and linear programming [69]. Note that the linear programming solution requires problem relaxation by allowing non-integer value for the association event and therefore the integrality of the solution is not ensured. This brings up the question of how to deal with the fractional assignments. In [69], the Lagrangian relaxation solution was shown to be better than the m -best assignment and linear programming implementations.

Recently, the MDA formulation of MHT was proven to be equivalent to the one finding the MWIS of a weighted undirected graph [70]. The MWIS formulation of MHT is easier not only to understand, but also to implement. However, the MWIS problem is still an NP-hard problem. To make it computationally tractable, one needs to seek to heuristic approaches or countable approximations. In [71], the authors utilised the MWIS formulation of MHT in the task of multi pedestrian tracking, where the MWIS problem was solved by a line search method. However, the quality of line search is highly dependent on the initial guess and hence easy to be trapped in a local optimum. Another MWIS implementation algorithm of MHT can be found in [72], where the exact solution [73] was sought at the price of increased computational demands. A better solution, in terms of the computational

power, to address the MWIS problem is the evolutionary algorithm [74] or heuristic approach [75].

Random Finite Set Filters

In the past few years, the random finite set (RFS) theory was found to be mathematically rigorous in formulating Bayesian MTT. The basic idea of RFS filters is to model target states and measurements as two RFSs and directly apply the Bayesian recursion to these set-valued random variables for track update. This process directly avoids data association and therefore RFS-based MTT algorithms usually are computationally efficient.

Typical examples of tractable RFS-based MTT algorithms are the probability hypothesis density (PHD) filter [76, 77, 78], the cardinalized probability hypothesis density (CPHD) filter [79], and the multi-target multi-Bernoulli filter [80]. The PHD (respectively, the CPHD) filter propagates the first moment of the multi-target density (respectively, both first and second moment). The limitation of the PHD filter is that its performance degrades drastically when the detection probability is low. As a comparison, the multi-target multi-Bernoulli filter directly approximates the multi-target probability density function by a multi-Bernoulli distribution. Although these RFS filters have great potentials in computational efficiency, they cannot infer the identity of each target and therefore have limited range of applications. For scenarios that target identification is not important, e.g., track-before-detect, the RFS filter is a wise option. The recently-proposed labelled RFS [81, 82] can handle the target identity problem, but the implementation requires a similar way as shown in classical data association approaches.

Summary of Existing Algorithms

The main characteristics of the aforementioned local MTT algorithms are summarised in Table 2.2. The advantages and disadvantages of the discussed distributed estimation algorithms are compared with respect to different metrics

- Robustness: the algorithm is robust against the environmental disturbances, e.g., clutter rate;
- Data association: whether or not the algorithm assessed requires data association in track update;

- Time complexity: the computational complexity of the algorithm;
- Target identification: whether or not the algorithm evaluated can provide the information on target identity.

In general, the performance of single-scan hard decision approaches, e.g., GNN, is not acceptable for scenarios with high clutter rate. For this reason, these approaches are not recommended for the applications using low-cost sensors. Since RFS filters have polynomial-time complexity and shows strong robustness against environmental disturbances, these filters have great potentials for low-cost sensors provided that the target identity is not important. When target identification is required for the considered missions, JPDA, MHT and labelled RFS are suitable choices. Among them, MHT provides the best performance if enough computational source is available. However, MHT runs in a sliding window and hence it is not a really real-time target tracking algorithm. Compared to MHT, both JPDA and labelled RFS provide reasonable performance with reduced computational overload.

Table 2.2: Characteristics of different local multi-target estimators.

Algorithm	Robustness	Data association	Time complexity	Target identification
NN [49], GNN [50]	Weak	Yes	Polynomial-time	Yes
JPDA [51, 53, 54, 55]	Strong	Yes	#P-complete	Yes
MHT [61, 68, 63, 72]	Strong	Yes	NP-hard	Yes
RFS [78, 79, 80]	Strong	No	Polynomial-time	No
Labelled RFS [81, 82]	Strong	Yes	#P-complete	Yes

Note that practical implementation of most MTT algorithms requires the full knowledge on the environment, i.e., target detection probability and clutter rate. Generally, the overall tracking performance is highly dependent on these two important parameters and they are usually empirically set in most applications. Significant mismatches of these two key parameters could result in unsatisfactory and even erroneous tracking results. Although the detection probability, dependent on the sensors and the scenarios, can be calibrated via offline experiments, it might not be cost-effective and they are typically unavailable for low-cost sensors. Additionally, sensors drift from their initial factory calibration during the lifetime or due to environmental changes, e.g., temperature, humidity, etc. Under dynamic environmental conditions or with different target detection approaches, the clutter rate generated by the same sensor might also change in a great deal: for example, in

image-based target tracking, measurements are extracted from various detection algorithms, ranging from simple background subtraction [83] to complicated deep learning approach [84]. Obviously, the clutter rate and detection probability are different for different detection methods. Therefore, how to accommodate the unknown detection probability and clutter rate is of critical importance in practice. Typical treatment of this problem is to decouple target detection probability or clutter rate estimation from the multi-target state estimation [85, 86, 87, 88]. These approaches utilise two separate filters and therefore cannot fully exploit the synergistic effect between. Recently, this issue is also tackled in RFS filters [89, 90], but they cannot preserve the labels of the targets and thus might have limited applications.

2.3.2 Distributed Multi-Target Fusion

After obtaining local MTT estimates from each sensor node, distributed MTT fusion can be performed to synergistically merge the redundant information and effectively complement the limitations of each sensor node by sharing information among locally connected sensors, hence improving the overall estimation quality. Unlike STT, distributed MTT requires two separate steps in the fusion stage:

- Track-to-track association to address the problem of target origin uncertainty;
- Information fusion considering data association uncertainty.

In a general distributed MTT mission, each sensor node orders its own local estimates and this information is unavailable to other sensors. In other words, the mappings of local estimates from two different sensors is unknown. Under this condition, track-to-track association is required to find the common source origin of local estimates that come from different sensors. Basically, track-to-track association is conceptually similar to measurement-to-track association in local MHT. For example, if we consider the first sensor as a reference sensor node, local estimates provided by other sensors can then be considered as virtual measurements that will be assigned to the local estimates generated by the first sensor node. With this mind, it can be concluded that the data association algorithms developed for MHT can be directly applied to address the problem of target origin uncertainty.

Since hard decision data association approaches, e.g., GNN and MHT, find the most likely measurement to update each target, the measurement update can be carried out using typical Kalman filter and its related algorithms, e.g., EKF, UKF, as long as the data association process is finished. This means that these local MTT algorithms

in updating one specific target share similar structure as traditional information-form Kalman filter. Therefore, the fusion strategies discussed in distributed STT can be applied to fuse local estimates after resolving the target origin uncertainty issue in track-to-track association. An example of this type of distributed MTT algorithm was reported in [90, 91], where the local MTT utilises GNN in data association and the fusion strategy is based on CMKF.

Unlike hard decision, utilising soft decision, i.e., JPDA, in local MTT introduces additional terms related to data association uncertainty in the fusion stage. By analysing the effect of data association uncertainty, the authors of [92] revealed that the centralised JPDA (CJPDA) for the i th target becomes

$$\begin{aligned} (\bar{\mathbf{P}}_{k|k}^i)^{-1} \mathbf{x}_{k|k}^i &= (\mathbf{P}_{k|k-1}^i)^{-1} \mathbf{x}_{k|k-1}^i + \sum_{l=1}^N \mathbf{g}_{k,l} \\ (\bar{\mathbf{P}}_{k|k}^i)^{-1} &= (\mathbf{P}_{k|k-1}^i)^{-1} + \sum_{l=1}^N \mathbf{H}_{k,l}^T \mathbf{R}_{k,l}^{-1} \mathbf{H}_{k,l} \\ (\mathbf{P}_{k|k}^i)^{-1} &= (\mathbf{P}_{k|k-1}^i)^{-1} + \sum_{l=1}^N \mathbf{G}_{k,l} \end{aligned} \quad (2.9)$$

The difference between Eq. (2.9) and Eq. (2.1) is resulted from the measurement origin uncertainty. This is reflected by the two information-related terms $\mathbf{g}_{k,l}$ and $\mathbf{G}_{k,l}$. Note that the centralised estimation (2.9) reduces to the original one, i.e., Eq. (2.1), if there is no data association uncertainty.

Following similar concept as KCF [19, 20], a distributed JPDA was developed in [93], which considers data association uncertainties when broadcasting information to local neighbours. This method, however, has similar drawbacks as KCF, i.e., requiring every sensor node and its neighbours have joint observability or at least detectability about the target of interest. Improved results were reported in [23, 94, 95] by combining JPDA filter with consensus on information weighted local estimates. For real implementation, this algorithm requires the global information on the network size, i.e., total number of sensors. In practice, an unexpected sensor failure will inevitably change the total number of nodes, leading to performance degradation if the original value of network size is used.

By incorporating consensus algorithm with RFS filters, e.g., PHD filter [96], CPHD filter [97, 98], distributed MTT algorithms were proposed without data association. These approaches, however, cannot preserve track continuity, e.g., no target identity information.

Apart from consensus, another mainstream in distributed multi-target tracking is sequentially fuse the information between two connected sensors [99, 100, 101]. Although this strategy is scalable, it requires each sensor's field-of-view to cover the entire surveillance region, which is not practical [100].

2.4 Conclusions

This chapter presents a critical and comprehensive review of several existing and recently-developed distributed fusion algorithms and compares their characteristics and performance in terms of different metrics. This provides readers deeper understanding about how to speedup capturing a given algorithm then comprehensive characterisations of that algorithm, how it complements other approaches, and how it can be integrated with them.

Chapter 3

Sample Greedy Gossip Distributed Kalman Filter

3.1 Introduction

It is known that leveraging proper sensor fusion algorithms over a sensor network can synergistically merge the redundant information and effectively complement the limitations of each sensor node, thus providing the possibility to improve the quality of local estimations beyond their current status [10, 102]. However, it follows from the discussions, presented in Chapter 2, that there is a strong tradeoff between communication cost and overall tracking performance. Therefore, the key of performing distributed estimation over a low-cost sensor network is how to well balance the communication overload and the estimation performance since low-cost sensors are generally battery powered with limited communication and computation abilities.

This chapter investigates the problem of distributed STT over a low-cost sensor network and proposes a new SGG-DKF algorithm to address this problem. The development of this new algorithm is motivated by the complementary characteristics of RG [36] and GG[103]. Specifically, RG has lower computational burden but its convergence rate is relatively slow because of the randomised nature. On the other hand, GG enjoys faster convergence to the average value at expense of higher communication burden. This can be attributed to the fact that GG requires each sensor node to communicate with all its connected neighbours to find an optimal path. Based on these observations, this chapter suggests a new variant of gossip-based DKF to exhibit positive features of both RG and GG. The promising feature

of the proposed SGG-DKF is that it offers great flexibility and well balance between communication cost and tracking accuracy. The main contributions of this chapter are twofold:

- A new SGG algorithm is proposed for distributed network-wide average computation. Unlike the RG, SGG takes the advantage of GG by applying the greedy node selection strategy to a randomly active sensor node set. As the expected size of the active node set is smaller than the number of locally-connected neighbours, SGG can reduce the communication overload in the average sense. Theoretical analysis reveals that the proposed SGG algorithm guarantees asymptotic convergence to the average state. The effect of the stochastic sampling strategy on the convergence rate of the proposed SGG algorithm is also theoretically analysed to demonstrate the tradeoff performance of SGG. To provide better insights into SGG, we derive the upper bound of its convergence rate, which relates the convergence performance of SGG to that of RG and GG.
- A new SGG-DKF is proposed for distributed target tracking over a large sensor network. The proposed SGG-DKF leverages the information weighted fusion rule while the previous gossip approaches utilised the concept of measurement vector fusion [39, 40]. This enables the algorithm developed to preserve the consistency in local estimates unlike the previous gossip algorithms. As the proposed SGG guarantees asymptotic convergence, the SGG-DKF ensures global convergence to the Bayesian optimal CKF. For realistic application with finite number of gossip iterations, we also theoretically prove that the fused covariance matrix is uniformly bounded.

3.2 Sample Greedy Gossip Algorithm

This section will propose a new SGG algorithm by exploiting the benefits of both RG [36] and GG [103]. We consider a network of N sensors and model the network topology as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \dots, N\}$ represents the sensor node set and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ denotes the edge set. If sensor nodes s and t can directly communicate with each other, then $(s, t) \in \mathcal{E}$. We assume that the sensor network is connected, e.g., any two sensor nodes can communicate with each other

through a multi-hop path¹. Denote a_s as the available information from the s th sensor and is initialised as $a_s(0)$. The objective of gossip algorithm is to enforce all local sensor nodes to make an agreement on the initial average value $\frac{1}{N} \sum_{s=1}^N a_s(0)$ using only local information available from the connected neighbours.

At the l th round of the RG iterations, a node $s \in \mathcal{V}$ is randomly selected for communication. This can be accomplished by using an asynchronous time mode², as described in [36]. Then, node s randomly selects a neighbour node $t \in \mathcal{N}_s$, where $\mathcal{N}_s$ represents the connected sensor nodes of the s th sensor (not including s), and performs information average as

$$a_s(l) = a_t(l) = \frac{a_s(l-1) + a_t(l-1)}{2} \quad (3.1)$$

For connected sensor network, it was shown in [36] that the RG algorithm guarantees asymptotic convergence to the initial average value as $\lim_{l \rightarrow \infty} a_s(l) = \frac{1}{N} \sum_{s=1}^N a_s(0)$. The main positive feature of RG is that it has relatively low communication burden since each sensor only needs to randomly communicate with one connected sensor node during one iteration round. However, the RG algorithm usually suffers from low convergence speed because of the randomised communication strategy, compared to the optimised deterministic communication path. For this reason, a GG algorithm was proposed in [103] to accelerate the convergence rate, i.e., reducing the number of required gossip iterations to make an agreement across the sensor nodes. The GG algorithm employs a deterministic procedure to select the communication nodes, e.g., active sensors. More specifically, at the l th round of the gossip iterations, the s th sensor selects a sensor node $t^* \in \mathcal{N}_s$ that has the largest information discrepancy as

$$t^* = \max_{t \in \mathcal{N}_s} [a_s(l) - a_t(l)]^2 \quad (3.2)$$

After finding t^* , the two sensors perform information averaging using Eq. (3.1). Compared with the original RG process, the GG algorithm is proved to provide improved convergence speed. However, this strategy requires each sensor node to communicate all connected sensor nodes to find the optimal communication path and therefore, might increase the communication overhead.

Motivated by the complementary properties of these two different gossip algorithms, this chapter develops a new variant of gossip algorithm, called SGG. The proposed

¹Note that this condition is required in guaranteeing the asymptotic convergence of gossip algorithms. However, gossip algorithms are also applicable to partially-connected sensor networks.

²Each node has a Poisson time clock with rate 1. At every gossip iteration, only one node is randomly picked up to communicate with one of its neighbours by clock ticking.

Algorithm 3.1 Sample greedy gossip

Input: Initial local information $a_s(0)$, maximum iteration step L , sensor activation probability p

Output: Fused information $a_s(L)$

```

1: Randomly selects a sensor node  $s$  from the sensor network
2: for  $l = 1 : L$  do
3:    $\mathcal{A}_s = \emptyset$  ▷ Initialise the active sensor node set  $\mathcal{A}_s$ 
4:   for  $i = 1 : |\mathcal{N}_s|$  do
5:     Sensor  $Sn(i)$  generates a sample  $q_i \sim \mathcal{U}(0, 1)$ 
6:     if  $q_i \geq 1 - p$  then
7:       Sensor  $Sn(i)$  decides to actively communicate with sensor  $s$ 
8:        $\mathcal{A}_s = \mathcal{A}_s \cup Sn(i)$  ▷ Update the active sensor node set  $\mathcal{A}_s$ 
9:     end if
10:  end for
11:  if  $\mathcal{A}_s \neq \emptyset$  then
12:     $t^* = \max_{t \in \mathcal{A}_s} [a_s(l-1) - a_t(l-1)]^2$  ▷ Greedy node selection from  $\mathcal{A}_s$ 
13:     $a_s(l) = a_{t^*}(l) = \frac{1}{2} [a_s(l-1) + a_{t^*}(l-1)]$ 
14:  else
15:    Randomly selects a sensor node  $t$  from  $\mathcal{N}_s$  ▷ Perform randomised gossip
16:     $a_s(l) = a_t(l) = \frac{1}{2} [a_s(l-1) + a_t(l-1)]$ 
17:  end if
18: end for

```

gossip algorithm enjoys the advantages of both RG and GG: relatively faster convergence speed and lower communication burden. The basic idea of SGG is to apply the greedy node selection strategy to a randomly active sensor node set $\mathcal{A}_s$, which is a subset of $\mathcal{N}_s$, i.e., $\mathcal{A}_s \in \mathcal{N}_s$. For notation convenience, we define $\mathcal{N}_s \triangleq \{Sn(1), Sn(2), \dots, Sn(|\mathcal{N}_s|)\}$ with $Sn(i) \in \{1, 2, \dots, N\}$ for all $i \in \{1, 2, \dots, |\mathcal{N}_s|\}$, where $|\mathcal{N}_s|$ stands for the cardinality of set $\mathcal{N}_s$. The active sensor node set $\mathcal{A}_s$ is generated by randomly selecting sensors from $\mathcal{N}_s$ for communication based on a stochastic uniform sampling procedure. More specifically, all sensors in $\mathcal{N}_s$ generate a sample, i.e., q_i for all $i \in \{1, 2, \dots, |\mathcal{N}_s|\}$, from uniform distribution $\mathcal{U}(0, 1)$. Each sensor from $\mathcal{N}_s$ is then activated with probability $p \in [0, 1]$: if $q_i \geq 1 - p$, sensor $Sn(i)$ decides to actively communicate with node s . If no sensor has been activated by the sampling strategy, i.e., $\mathcal{A}_s = \emptyset$, we perform RG for update. The pseudo code of the proposed SGG algorithm is summarised in Algorithm 3.1.

Remark 3.1. *It can be clearly noted that the proposed algorithm only utilises a random subset of $\mathcal{N}_s$ to perform greedy sensor node selection. The communication burden, therefore, depends on the sensor activation probability p , which determines the size of $\mathcal{A}_s$. For example, if the node activation probability p is chosen as $p = 0.5$, the communication burden is half in average sense at every gossip iteration, compared*

with the GG. Therefore, the proposed SGG can be viewed as a generalised version of the gossip algorithm that can trade-off between RG and GG: the proposed SGG becomes GG with $p = 1$ and reduces to RG with $p = 0$.

3.3 Analysis of Sample Greedy Gossip

In this section, we provide detailed theoretical analysis of the proposed SGG algorithm, demonstrating its convergence performance and comparing its convergence rate with that of RG and GG. Note that the analysis carried out in this section is based on the asynchronous time mode. However, the extension to synchronous time model³ is straightforward and the qualitative as well as quantitative results are unaffected by the type of time model [104, 105, 36].

3.3.1 Asymptotic Convergence Analysis

This subsection analyses the asymptotic convergence of the proposed SGG algorithm. For notation convenience, we denote $\mathbf{a}(l) = [a_1(l), a_2(l), \dots, a_N(l)]^T$ and define $\bar{\mathbf{a}}$ as a column vector with each element being $\frac{1}{N} \sum_{l=1}^N a_l(0)$. The main results are presented in the following theorem.

Theorem 3.1. *The proposed SGG algorithm guarantees asymptotic convergence to the average state $\bar{\mathbf{a}}$.*

Proof. Assume that the s th and the t th sensors perform gossip at the l th iteration of SGG, then the recursive update of SGG can be obtained as

$$\mathbf{a}(l) = \mathbf{a}(l-1) - \frac{1}{2}\mathbf{g}(l) \quad (3.3)$$

where $\mathbf{g}(l) \in \mathbb{R}^N$ is a column vector with its elements being

$$g_i(l) = \begin{cases} a_s(l-1) - a_t(l-1), & \text{for } i = s \\ -(a_s(l-1) - a_t(l-1)), & \text{for } i = t \\ 0, & \text{otherwise} \end{cases} \quad (3.4)$$

³Unlike asynchronous time model, all nodes in every gossip iteration communicate simultaneously with one of their neighbour.

Based on Eq. (3.3), the recursive update of the squared error is determined as

$$\begin{aligned}
\|\mathbf{a}(l) - \bar{\mathbf{a}}\|^2 &= \left\| \mathbf{a}(l-1) - \frac{1}{2}\mathbf{g}(l) - \bar{\mathbf{a}} \right\|^2 \\
&= \|\mathbf{a}(l-1) - \bar{\mathbf{a}}\|^2 - [\mathbf{a}(l-1) - \bar{\mathbf{a}}]^T \mathbf{g}(l) + \frac{1}{4}\|\mathbf{g}(l)\|^2 \\
&= \|\mathbf{a}(l-1) - \bar{\mathbf{a}}\|^2 - \frac{1}{2}[a_s(l-1) - a_t(l-1)]^2
\end{aligned} \tag{3.5}$$

Note that both s and t are random in the proposed SGG. For this reason, we will examine the expected squared error, i.e., $\mathbb{E}[\|\mathbf{a}(l) - \bar{\mathbf{a}}\|^2]$, in the following analysis. Taking the expectation on $[a_s(l-1) - a_t(l-1)]^2$ gives

$$\begin{aligned}
&\mathbb{E}\{[a_s(l-1) - a_t(l-1)]^2\} \\
&= \frac{1}{N} \sum_{s=1}^N \sum_{m=1}^{|\mathcal{N}_s|} p^m (1-p)^{|\mathcal{N}_s|-m} \sum_{\mathcal{N}_{s,m} \in \{\mathcal{N}_{s,m}\}} \max_{t \in \mathcal{N}_{s,m}} [a_s(l-1) - a_t(l-1)]^2 \\
&\quad + \frac{1}{N} \sum_{s=1}^N (1-p)^{|\mathcal{N}_s|} \frac{1}{|\mathcal{N}_s|} \sum_{t \in \mathcal{N}_s} [a_s(l-1) - a_t(l-1)]^2
\end{aligned} \tag{3.6}$$

where $\mathcal{N}_{s,m}$ denotes a set of m sensor nodes, randomly drawn from $\mathcal{N}_s$ and $\{\mathcal{N}_{s,m}\}$ stands for the set that includes all possible $\mathcal{N}_{s,m}$.

Note that the first term on the right hand side of Eq. (3.6) refers to $1 \leq m \leq |\mathcal{N}_s|$ sensors from $\mathcal{N}_s$ determine to communicate with the s th sensor, and the second term stands for no sensor has been activated by the sampling procedure. As stated in Algorithm 3.1, if no sensor decides to communicate with node s during the sampling phase, we perform the RG for update. From Eq. (3.6), it is clear that $\mathbb{E}\{[a_s(l-1) - a_t(l-1)]^2\} \geq 0$, where the equality holds if and only if $\mathbf{a}(l-1) = \bar{\mathbf{a}}$. This means that, unless all nodes make agreement on the average state $\bar{\mathbf{a}}$, the proposed SGG algorithm will make progress in expectation towards the average state $\bar{\mathbf{a}}$. Moreover, by repeatedly applying recursion (3.6) and taking the expectation, we have

$$\begin{aligned}
\mathbb{E}[\|\mathbf{a}(l) - \bar{\mathbf{a}}\|^2] &= \mathbb{E}[\|\mathbf{a}(l-1) - \bar{\mathbf{a}}\|^2] - \frac{1}{2}\mathbb{E}\{[a_s(l-1) - a_t(l-1)]^2\} \\
&= \mathbb{E}[\|\mathbf{a}(0) - \bar{\mathbf{a}}\|^2] - \frac{1}{2} \sum_{i=1}^l \mathbb{E}\{[a_s(i-1) - a_t(i-1)]^2\}
\end{aligned} \tag{3.7}$$

Since $\mathbb{E} [\|\mathbf{a}(l) - \bar{\mathbf{a}}\|^2] \geq 0$, we have

$$\mathbb{E} [\|\mathbf{a}(0) - \bar{\mathbf{a}}\|^2] \geq \frac{1}{2} \sum_{i=1}^l \mathbb{E} \{[a_s(i-1) - a_t(i-1)]^2\} \quad (3.8)$$

which implies that $\mathbb{E} \{[a_s(l-1) - a_t(l-1)]^2\} \rightarrow 0$ as $l \rightarrow \infty$. As the solution $\mathbf{a}(l) = \bar{\mathbf{a}}$ is the only stationary point of stochastic recursion (3.5), the proposed SGG algorithm guarantees asymptotic convergence to the average state $\bar{\mathbf{a}}$. $\square$

3.3.2 Convergence Rate Analysis

This subsection analyses convergence rate of the proposed SGG algorithm. We first investigates the effect of sensor activation probability p on the convergence speed and then derive the convergence bound.

Effect of Sensor Activation Probability

The tradeoff performance of the proposed SGG algorithm highly depends on the sensor activation probability p . For this reason, we will theoretically analyse the effect of sensor activation probability p and the result is presented in the following theorem.

Theorem 3.2. *The convergence rate of the proposed SGG algorithm increases with higher sensor activation probability.*

Proof. Define

$$\begin{aligned} f_s(p) = & \sum_{m=1}^{|\mathcal{N}_s|} p^m (1-p)^{|\mathcal{N}_s|-m} \sum_{\mathcal{N}_{s,m} \in \{\mathcal{N}_{s,m}\}} \max_{t \in \mathcal{N}_{s,m}} [a_s(l-1) - a_t(l-1)]^2 \\ & + (1-p)^{|\mathcal{N}_s|} \frac{1}{|\mathcal{N}_s|} \sum_{t \in \mathcal{N}_s} [a_s(l-1) - a_t(l-1)]^2 \end{aligned} \quad (3.9)$$

From Eqs. (3.7) and (3.6), it is clear that the expected reduction of the square is given by

$$\mathbb{E} [\|\mathbf{a}(l-1) - \bar{\mathbf{a}}\|^2] - \mathbb{E} [\|\mathbf{a}(l) - \bar{\mathbf{a}}\|^2] = \frac{1}{2N} \sum_{s=1}^N f_s(p) \quad (3.10)$$

Taking the derivative of $f_s(p)$ with respect to p gives

$$\begin{aligned} \frac{\partial f_s(p)}{\partial p} &= \sum_{m=1}^{|\mathcal{N}_s|} \frac{\partial p^m (1-p)^{|\mathcal{N}_s|-m}}{\partial p} \sum_{\mathcal{N}_{s,m} \in \{\mathcal{N}_{s,m}\}} \max_{t \in \mathcal{N}_{s,m}} [a_s(l-1) - a_t(l-1)]^2 \\ &\quad + \frac{\partial(1-p)^{|\mathcal{N}_s|}}{\partial p} \frac{1}{|\mathcal{N}_s|} \sum_{t \in \mathcal{N}_s} [a_s(l-1) - a_t(l-1)]^2 \end{aligned} \quad (3.11)$$

For a specific m , the number of possible $\mathcal{N}_{s,m}$ is given by

$$C_{|\mathcal{N}_s|}^m = \frac{|\mathcal{N}_s|!}{m! (|\mathcal{N}_s| - m)!} \quad (3.12)$$

Now let us define $m^* = \lfloor |\mathcal{N}_s| p \rfloor$. Then, it is trivial that

$$\begin{aligned} (m - |\mathcal{N}_s| p) \sum_{\mathcal{N}_{s,m} \in \{\mathcal{N}_{s,m}\}} \max_{t \in \mathcal{N}_{s,m}} [a_s(l-1) - a_t(l-1)]^2 \\ \geq (m - |\mathcal{N}_s| p) \frac{C_{|\mathcal{N}_s|}^m}{C_{|\mathcal{N}_s|}^{m^*}} \sum_{\mathcal{N}_{s,m^*} \in \{\mathcal{N}_{s,m^*}\}} \max_{t \in \mathcal{N}_{s,m^*}} [a_s(l-1) - a_t(l-1)]^2 \end{aligned} \quad (3.13)$$

$$\begin{aligned} \sum_{t \in \mathcal{N}_s} [a_s(l-1) - a_t(l-1)]^2 \\ \leq \frac{|\mathcal{N}_s|}{C_{|\mathcal{N}_s|}^{m^*}} \sum_{\mathcal{N}_{s,m^*} \in \{\mathcal{N}_{s,m^*}\}} \max_{t \in \mathcal{N}_{s,m^*}} [a_s(l-1) - a_t(l-1)]^2 \end{aligned} \quad (3.14)$$

Substituting Eqs. (3.13) and (3.14) into Eq. (3.11) yields

$$\begin{aligned} \frac{\partial f_s(p)}{\partial p} &\geq \sum_{m=1}^{|\mathcal{N}_s|} p^{m-1} (1-p)^{|\mathcal{N}_s|-m-1} (m - |\mathcal{N}_s| p) \frac{C_{|\mathcal{N}_s|}^m}{C_{|\mathcal{N}_s|}^{m^*}} \\ &\quad \times \sum_{\mathcal{N}_{s,m^*} \in \{\mathcal{N}_{s,m^*}\}} \max_{t \in \mathcal{N}_{s,m^*}} [a_s(l-1) - a_t(l-1)]^2 \\ &\quad - (1-p)^{|\mathcal{N}_s|-1} \frac{|\mathcal{N}_s|}{C_{|\mathcal{N}_s|}^{m^*}} \sum_{\mathcal{N}_{s,m^*} \in \{\mathcal{N}_{s,m^*}\}} \max_{t \in \mathcal{N}_{s,m^*}} [a_s(l-1) - a_t(l-1)]^2 \\ &= \sum_{m=0}^{|\mathcal{N}_s|} m p^m (1-p)^{|\mathcal{N}_s|-m} \frac{C_{|\mathcal{N}_s|}^m}{p(1-p) C_{|\mathcal{N}_s|}^{m^*}} \sum_{\mathcal{N}_{s,m^*} \in \{\mathcal{N}_{s,m^*}\}} \max_{t \in \mathcal{N}_{s,m^*}} [a_s(l-1) - a_t(l-1)]^2 \\ &\quad - \sum_{m=0}^{|\mathcal{N}_s|} |\mathcal{N}_s| p^m (1-p)^{|\mathcal{N}_s|-m} \frac{C_{|\mathcal{N}_s|}^m}{(1-p) C_{|\mathcal{N}_s|}^{m^*}} \sum_{\mathcal{N}_{s,m^*} \in \{\mathcal{N}_{s,m^*}\}} \max_{t \in \mathcal{N}_{s,m^*}} [a_s(l-1) - a_t(l-1)]^2 \end{aligned} \quad (3.15)$$

Based on the property of binomial distribution, it can be concluded that

$$\begin{aligned} \sum_{m=0}^{|\mathcal{N}_s|} m C_{|\mathcal{N}_s|}^m p^m (1-p)^{|\mathcal{N}_s|-m} &= |\mathcal{N}_s| p \\ \sum_{m=0}^{|\mathcal{N}_s|} C_{|\mathcal{N}_s|}^m p^m (1-p)^{|\mathcal{N}_s|-m} &= 1 \end{aligned} \quad (3.16)$$

Then, evaluating Eq. (3.15) using Eq. (3.16) gives

$$\frac{\partial f_s(p)}{\partial p} \geq 0 \quad (3.17)$$

where the equality holds if and only if $\mathbf{a}(l-1) = \bar{\mathbf{a}}$. From Eqs. (3.10) and (3.17), we have

$$\frac{\partial (\mathbb{E} [\|a(l-1) - \bar{a}\|^2] - \mathbb{E} [\|a(l) - \bar{a}\|^2])}{\partial p} = \frac{1}{2N} \sum_{s=1}^N \frac{\partial f_s(p)}{\partial p} \geq 0 \quad (3.18)$$

where the equality again holds if and only if $\mathbf{a}(l-1) = \bar{\mathbf{a}}$. $\square$

Remark 3.2. *Theorem 3.2 demonstrates that the proposed SGG algorithm provides faster convergence rate with higher sensor activation probability. This clearly reveals that SGG is a tradeoff between RG ($p = 0$) and GG ($p = 1$). In viewing of this fact, it can be concluded that the proposed SGG algorithm provides great flexibility and well balance between communication cost and convergence performance. If enough resource is available for communication, then a higher sensor activation probability can be chosen to increase the convergence rate; otherwise, a relatively small value of p is desirable.*

Convergence Bound

To show the tradeoff convergence performance of the proposed SGG algorithm, we assume that the sensor activation probability satisfies $p \in (0, 1)$ in the following analysis. To analyse the bound on the convergence rate, we will first derive the lower and upper convergence bounds of each SGG iteration. The result is established in the following lemma.

Lemma 3.1. *The expected reduction of the square error of each SGG iteration is*

lower and upper bounded by

$$\begin{aligned} \mathbb{E} [\|\mathbf{a}(l-1) - \bar{\mathbf{a}}\|^2] - \mathbb{E} [\|\mathbf{a}(l) - \bar{\mathbf{a}}\|^2] \\ \geq \frac{1}{2N} \sum_{s=1}^N \frac{1}{|\mathcal{N}_s|} \sum_{t \in \mathcal{N}_s} [a_s(l-1) - a_t(l-1)]^2 \end{aligned} \quad (3.19)$$

$$\begin{aligned} \mathbb{E} [\|\mathbf{a}(l-1) - \bar{\mathbf{a}}\|^2] - \mathbb{E} [\|\mathbf{a}(l) - \bar{\mathbf{a}}\|^2] \\ \leq \frac{1}{2N} \sum_{s=1}^N \max_{t \in \mathcal{N}_s} [a_s(l-1) - a_t(l-1)]^2 \end{aligned} \quad (3.20)$$

where the equality holds if and only if $\mathbf{a}(l-1) = \bar{\mathbf{a}}$.

Proof. Note that the probability that $\mathcal{N}_{s,m}$ contains one specific sensor node $t \in \mathcal{N}_s$ is

$$\Pr(t \in \mathcal{N}_s \cup \mathcal{N}_{s,m}) = \frac{m}{|\mathcal{N}_s|} \quad (3.21)$$

Let $M_{s,m}^t$ be the number of occurrences of one specific sensor node $t \in \mathcal{N}_s$ that appears in $\{\mathcal{N}_{s,m}\}^4$, which is given by

$$\begin{aligned} M_{s,m}^t &= \Pr(t \in \mathcal{N}_s \cup \mathcal{N}_{s,m}) C_{|\mathcal{N}_s|}^m \\ &= \frac{(|\mathcal{N}_s| - 1)!}{(m-1)! (|\mathcal{N}_s| - m)!} \end{aligned} \quad (3.22)$$

Using Eq. (3.22), we obtain

$$\begin{aligned} \sum_{\mathcal{N}_{s,m} \in \{\mathcal{N}_{s,m}\}} \max_{t \in \mathcal{N}_{s,m}} [a_s(l-1) - a_t(l-1)]^2 \\ \geq \sum_{\mathcal{N}_{s,m} \in \{\mathcal{N}_{s,m}\}} \frac{1}{m} \sum_{t \in \mathcal{N}_{s,m}} [a_s(l-1) - a_t(l-1)]^2 \\ = \frac{1}{m} \sum_{t \in \mathcal{N}_s} M_{s,m}^t [a_s(l-1) - a_t(l-1)]^2 \\ = \frac{1}{|\mathcal{N}_s|} C_{|\mathcal{N}_s|}^m \sum_{t \in \mathcal{N}_s} [a_s(l-1) - a_t(l-1)]^2 \end{aligned} \quad (3.23)$$

⁴Here, we provide an example to explain the notation $M_{s,m}^t$. Consider $\mathcal{N}_s = \{1, 2, 3, 4\}$ and $m = 2$. From the definition of $\mathcal{N}_{s,m}$, it can be concluded that $\mathcal{N}_{s,2}$ in this example has $C_4^2 = 6$ possibilities, i.e., $\{1, 2\}$, $\{1, 3\}$, $\{1, 4\}$, $\{2, 3\}$, $\{2, 4\}$, $\{3, 4\}$. Therefore, we have $\{\mathcal{N}_{s,2}\} = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}$ and the number of occurrences of each sensor index in $\{\mathcal{N}_{s,2}\}$ is given by $M_{s,2}^1 = M_{s,2}^2 = M_{s,2}^3 = M_{s,2}^4 = \frac{2}{4} C_4^2 = 3$.

Substituting Eq. (3.23) into Eq. (3.6) yields

$$\begin{aligned}
& \mathbb{E} \{ [a_s(l-1) - a_t(l-1)]^2 \} \\
& \geq \frac{1}{N} \sum_{s=1}^N \sum_{m=1}^{|\mathcal{N}_s|} C_{|\mathcal{N}_s|}^m p^m (1-p)^{|\mathcal{N}_s|-m} \frac{1}{|\mathcal{N}_s|} \sum_{t \in \mathcal{N}_s} [a_s(l-1) - a_t(l-1)]^2 \\
& \quad + \frac{1}{N} \sum_{s=1}^N (1-p)^{|\mathcal{N}_s|} \frac{1}{|\mathcal{N}_s|} \sum_{t \in \mathcal{N}_s} [a_s(l-1) - a_t(l-1)]^2 \\
& = \frac{1}{N} \sum_{s=1}^N \frac{1}{|\mathcal{N}_s|} \sum_{t \in \mathcal{N}_s} [a_s(l-1) - a_t(l-1)]^2 \sum_{m=0}^{|\mathcal{N}_s|} C_{|\mathcal{N}_s|}^m p^m (1-p)^{|\mathcal{N}_s|-m} \\
& = \frac{1}{N} \sum_{s=1}^N \frac{1}{|\mathcal{N}_s|} \sum_{t \in \mathcal{N}_s} [a_s(l-1) - a_t(l-1)]^2
\end{aligned} \tag{3.24}$$

where the equality holds if and only if $\mathbf{a}(l-1) = \bar{\mathbf{a}}$.

Also, $\mathbb{E} \{ [a_s(l-1) - a_t(l-1)]^2 \}$ can be upper bounded by

$$\begin{aligned}
& \mathbb{E} \{ [a_s(l-1) - a_t(l-1)]^2 \} \\
& \leq \frac{1}{N} \sum_{s=1}^N \sum_{m=1}^{|\mathcal{N}_s|} C_{|\mathcal{N}_s|}^m p^m (1-p)^{|\mathcal{N}_s|-m} \max_{t \in \mathcal{N}_s} [a_s(l-1) - a_t(l-1)]^2 \\
& \quad + \frac{1}{N} \sum_{s=1}^N (1-p)^{|\mathcal{N}_s|} \max_{t \in \mathcal{N}_s} [a_s(l-1) - a_t(l-1)]^2 \\
& = \frac{1}{N} \sum_{s=1}^N \max_{t \in \mathcal{N}_s} [a_s(l-1) - a_t(l-1)]^2 \sum_{m=0}^{|\mathcal{N}_s|} C_{|\mathcal{N}_s|}^m p^m (1-p)^{|\mathcal{N}_s|-m} \\
& = \frac{1}{N} \sum_{s=1}^N \max_{t \in \mathcal{N}_s} [a_s(l-1) - a_t(l-1)]^2
\end{aligned} \tag{3.25}$$

where the equality holds if and only if $\mathbf{a}(l-1) = \bar{\mathbf{a}}$. Combining the preceding two equations leads to the proof. $\square$

Remark 3.3. Note that the right hand side of inequality (3.19) is the expected reduction of square error of RG and the right hand side of inequality (3.20) is the expected reduction of square error of GG. Lemma 3.1 relates the performance of the proposed SGG to that of RG and GG. From Lemma 3.1, it is clear that the expected reduction of the square error of the proposed SGG algorithm is lower bounded by the RG and upper bounded by the GG. With this in mind, it can be concluded that the convergence rate of SGG is faster than the RG but slower than the GG in expectation.

Now, let us directly examine the convergence speed of the proposed SGG algorithm in terms of the ϵ -average time [36],

$$T_{ave}(\epsilon) = \sup_{\mathbf{a}(0) \neq \mathbf{0}} \inf \left\{ l : \Pr \left(\frac{\|\mathbf{a}(l) - \bar{\mathbf{a}}\|}{\|\mathbf{a}(0) - \bar{\mathbf{a}}\|} \geq \epsilon \right) \leq \epsilon \right\} \quad (3.26)$$

For notation convenience, let $\mathbf{W}^{SGG}(l)$ be the update transition matrix of implementing SGG at the l th iteration of SGG, e.g., $\mathbf{a}(l) = \mathbf{W}^{SGG}(l) \mathbf{a}(l-1)$, and $\mathbf{W}^{SGG}(1:l) = \prod_{j=1}^l \mathbf{W}^{SGG}(j)$ be the successive application of l SGG updates. Likewise, we denote the one-step update matrices of RG as $\mathbf{W}^{RG}(l)$; and define the application of l successive RG updates as $\mathbf{W}^{RG}(1:l) = \prod_{j=1}^l \mathbf{W}^{RG}(j)$. We further denote $\bar{\mathbf{W}} = \mathbb{E}[\mathbf{W}^{RG}(l)]$ as the expected value of the RG one-step update matrix, and let $\lambda_2(\bar{\mathbf{W}})$ be the second largest eigenvalue of $\bar{\mathbf{W}}$. The following theorem establishes the theoretical upper bound of ϵ -average time of the proposed SGG algorithm.

Theorem 3.3. *The ϵ -average time of the proposed SGG algorithm is upper bounded by*

$$T_{ave}^{SGG}(\epsilon) \leq \frac{3 \log \epsilon^{-1}}{\log \left(\lambda_2(\bar{\mathbf{W}}) - \min_{i \in [l]} \{\beta_i\} \right)^{-1}} \quad (3.27)$$

where

$$\beta_l = \frac{\eta_l}{\mathbb{E}[\|\mathbf{W}^{SGG}(1:l-1)\mathbf{a}(0) - \bar{\mathbf{a}}\|^2]} \geq 0 \quad (3.28)$$

with $\eta_l = 0$ if $\mathbf{a}(l-1) = \bar{\mathbf{a}}$, and otherwise,

$$\begin{aligned} \eta_l = & \frac{1}{2N} \sum_{s=1}^N \sum_{m=1}^{|\mathcal{N}_s|} p^m (1-p)^{|\mathcal{N}_s|-m} \sum_{\mathcal{N}_{s,m} \in \{\mathcal{N}_{s,m}\}} \max_{t \in \mathcal{N}_{s,m}} [a_s(l-1) - a_t(l-1)]^2 \\ & + \frac{1}{2N} \sum_{s=1}^N (1-p)^{|\mathcal{N}_s|} \frac{1}{|\mathcal{N}_s|} \sum_{t \in \mathcal{N}_s} [a_s(l-1) - a_t(l-1)]^2 \\ & - \frac{1}{2N} \sum_{s=1}^n \frac{1}{|\mathcal{N}_s|} \sum_{t \in \mathcal{N}_s} (a_s(l-1) - a_t(l-1))^2 \end{aligned} \quad (3.29)$$

Proof. The recursive reduction of the expected square error of RG satisfies [36]

$$\begin{aligned} \mathbb{E}[\|\mathbf{W}^{RG}(1:l)\mathbf{a}(0) - \bar{\mathbf{a}}\|^2] &= \mathbb{E}[\|\mathbf{W}^{RG}(1:l-1)\mathbf{a}(0) - \bar{\mathbf{a}}\|^2] \\ &\quad - \frac{1}{2N} \sum_{s=1}^n \frac{1}{|\mathcal{N}_s|} \sum_{t \in \mathcal{N}_s} (a_s(l-1) - a_t(l-1))^2 \end{aligned} \quad (3.30)$$

and is upper bounded by

$$\mathbb{E} \left[\left\| \mathbf{W}^{RG}(1:l) \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right] \leq \lambda_2(\bar{\mathbf{W}}) \mathbb{E} \left[\left\| \mathbf{W}^{RG}(1:l-1) \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right] \quad (3.31)$$

By implementing one step RG after applying $l-1$ steps of SGG updates and using the relationship of Eqs. (3.30) and (3.31), we have

$$\begin{aligned} \mathbb{E} \left[\left\| \mathbf{W}^{RG}(k) \mathbf{W}^{SGG}(1:l-1) \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right] &= \mathbb{E} \left[\left\| \mathbf{W}^{SGG}(1:l-1) \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right] \\ &\quad - \frac{1}{2N} \sum_{s=1}^n \frac{1}{|\mathcal{N}_s|} \sum_{t \in \mathcal{N}_s} (a_s(l-1) - a_t(l-1))^2 \\ &\leq \lambda_2(\bar{\mathbf{W}}) \mathbb{E} \left[\left\| \mathbf{W}^{SGG}(1:l-1) \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right] \end{aligned} \quad (3.32)$$

Using Eq. (3.32), the convergence of the proposed SGG can be obtained as

$$\begin{aligned} &\mathbb{E} \left[\left\| \mathbf{W}^{SGG}(1:l) \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right] \\ &= \mathbb{E} \left[\left\| \mathbf{W}^{SGG}(1:l-1) \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right] - \frac{1}{2N} \sum_{s=1}^n \frac{1}{|\mathcal{N}_s|} \sum_{t \in \mathcal{N}_s} (a_s(l-1) - a_t(l-1))^2 \\ &\quad - \frac{1}{2N} \sum_{s=1}^N \sum_{m=1}^{|\mathcal{N}_s|} p^m (1-p)^{|\mathcal{N}_s|-m} \sum_{\mathcal{N}_{s,m} \in \{\mathcal{N}_{s,m}\}} \max_{t \in \mathcal{N}_{s,m}} [a_s(l-1) - a_t(l-1)]^2 \\ &\quad - \frac{1}{2N} \sum_{s=1}^N (1-p)^{|\mathcal{N}_s|} \frac{1}{|\mathcal{N}_s|} \sum_{t \in \mathcal{N}_s} [a_s(l-1) - a_t(l-1)]^2 \\ &\quad + \frac{1}{2N} \sum_{s=1}^n \frac{1}{|\mathcal{N}_s|} \sum_{t \in \mathcal{N}_s} (a_s(l-1) - a_t(l-1))^2 \\ &= \mathbb{E} \left[\left\| \mathbf{W}^{SGG}(1:l-1) \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right] - \frac{1}{2N} \sum_{s=1}^n \frac{1}{|\mathcal{N}_s|} \sum_{t \in \mathcal{N}_s} (a_s(l-1) - a_t(l-1))^2 - \eta_l \end{aligned} \quad (3.33)$$

According to Lemma 3.1, it is clear that $\eta_l \geq 0$ and the equality holds if and only if $\mathbf{a}(l-1) = \bar{\mathbf{a}}$. With this in mind, substituting Eq. (3.32) into Eq. (3.33) gives the convergence bound of the proposed SGG as

$$\begin{aligned} &\mathbb{E} \left[\left\| \mathbf{W}^{SGG}(1:l) \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right] \\ &\leq [\lambda_2(\bar{\mathbf{W}}) - \beta_l] \mathbb{E} \left[\left\| \mathbf{W}^{SGG}(1:l-1) \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right] \end{aligned} \quad (3.34)$$

Repeatedly using Eq. (3.34) yields

$$\begin{aligned} \mathbb{E} \left[\left\| \mathbf{W}^{SGG}(1:l) \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right] &\leq \left\| \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \prod_{i=1}^l (\lambda_2(\bar{\mathbf{W}}) - \beta_i) \\ &\leq \left\| \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \left(\lambda_2(\bar{\mathbf{W}}) - \min_{i \in [l]} \{\beta_i\} \right)^l \end{aligned} \quad (3.35)$$

where the equality holds if and only if $\mathbf{a}(l-1) = \bar{\mathbf{a}}$.

Using Eq. (3.35) with the help of Markov's inequality, we have

$$\begin{aligned} \Pr \left(\frac{\left\| \mathbf{W}^{SGG}(1:l) \mathbf{a}(0) - \bar{\mathbf{a}} \right\|}{\left\| \mathbf{a}(0) - \bar{\mathbf{a}} \right\|} \geq \epsilon \right) &= \Pr \left(\frac{\left\| \mathbf{W}^{SGG}(1:l) \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2}{\left\| \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2} \geq \epsilon^2 \right) \\ &\leq \frac{\mathbb{E} \left[\left\| \mathbf{W}^{SGG}(1:l) \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right]}{\epsilon^2 \left\| \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2} \\ &\leq \epsilon^{-2} \left(\lambda_2(\bar{\mathbf{W}}) - \min_{i \in [l]} \{\beta_i\} \right)^l \end{aligned} \quad (3.36)$$

Based on Eq. (3.26), one can imply that the ϵ -averaging time for SGG is upper bounded by

$$T_{ave}^{SGG}(\epsilon) \leq \frac{3 \log \epsilon^{-1}}{\log \left(\lambda_2(\bar{\mathbf{W}}) - \min_{i \in [l]} \{\beta_i\} \right)^{-1}} \quad (3.37)$$

which completes the proof. $\square$

Remark 3.4. *Theorem 3.3 provides a theoretical upper bound for the ϵ -averaging time of the proposed SGG algorithm. Note from Eqs. (3.9) and (3.29) that η_l and $f_s(p)$ have the same monotonicity with respect to p . With this in mind, it can be easily verified that the upper bound of the ϵ -averaging time becomes smaller with higher sensor activation probability.*

3.3.3 Convergence Rate Comparison

This subsection will characterise the upper bound of ϵ -averaging time of SGG algorithm with respect to other gossip algorithms to provide better understandings and insights of the proposed algorithm. Similar to previous subsection, we denote the one-step update matrices of GG as $\mathbf{W}^{GG}(l)$; and define the application of l successive GG updates as $\mathbf{W}^{GG}(1:l) = \prod_{j=1}^l \mathbf{W}^{GG}(j)$. The main results of this subsection are provided in the following theorem.

Theorem 3.4. *The upper bound of ϵ -average time, denoted as $\mathcal{U}(\epsilon)$, of the proposed SGG algorithm is smaller than that of the RG, but larger than that of the GG, i.e.,*

$$\mathcal{U}_{GG}(\epsilon) \leq \mathcal{U}_{SGG}(\epsilon) \leq \mathcal{U}_{RG}(\epsilon) \quad (3.38)$$

Proof. Similarly, by implementing one step SGG after applying $l - 1$ steps of GG updates and using Eq. (3.34), we have

$$\begin{aligned} \mathbb{E} \left[\left\| \mathbf{W}^{SGG}(l) \mathbf{W}^{GG}(1 : l - 1) \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right] &= \mathbb{E} \left[\left\| \mathbf{W}^{GG}(1 : l - 1) \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right] \\ &\quad - \frac{1}{2N} \sum_{s=1}^N \sum_{m=1}^{|\mathcal{N}_s|} p^m (1-p)^{|\mathcal{N}_s|-m} \sum_{\mathcal{N}_{s,m} \in \{\mathcal{N}_{s,m}\}} \max_{t \in \mathcal{N}_{s,m}} [a_s(l-1) - a_t(l-1)]^2 \\ &\quad - \frac{1}{2N} \sum_{s=1}^n \frac{1}{|\mathcal{N}_s|} \sum_{t \in \mathcal{N}_s} (a_s(l-1) - a_t(l-1))^2 \\ &\leq [\lambda_2(\bar{\mathbf{W}}) - \beta_l] \mathbb{E} \left[\left\| \mathbf{W}^{GG}(1 : l - 1) \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right] \end{aligned} \quad (3.39)$$

Note that the recursive reduction of the expected square error of GG satisfies [103]

$$\begin{aligned} \mathbb{E} \left[\left\| \mathbf{W}^{GG}(1 : l) \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right] &= \mathbb{E} \left[\left\| \mathbf{W}^{GG}(1 : l - 1) \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right] \\ &\quad - \frac{1}{2N} \sum_{s=1}^N \max_{t \in \mathcal{N}_s} [a_s(l-1) - a_t(l-1)]^2 \end{aligned} \quad (3.40)$$

which is equivalent to

$$\begin{aligned} \mathbb{E} \left[\left\| \mathbf{W}^{GG}(1 : l) \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right] &= \mathbb{E} \left[\left\| \mathbf{W}^{GG}(1 : l - 1) \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right] \\ &\quad - \frac{1}{2N} \sum_{s=1}^N \sum_{m=1}^{|\mathcal{N}_s|} p^m (1-p)^{|\mathcal{N}_s|-m} \sum_{\mathcal{N}_{s,m} \in \{\mathcal{N}_{s,m}\}} \max_{t \in \mathcal{N}_{s,m}} [a_s(l-1) - a_t(l-1)]^2 \\ &\quad - \frac{1}{2N} \sum_{s=1}^N (1-p)^{|\mathcal{N}_s|} \frac{1}{|\mathcal{N}_s|} \sum_{t \in \mathcal{N}_s} [a_s(l-1) - a_t(l-1)]^2 \\ &\quad + \frac{1}{2N} \sum_{s=1}^N \sum_{m=1}^{|\mathcal{N}_s|} p^m (1-p)^{|\mathcal{N}_s|-m} \sum_{\mathcal{N}_{s,m} \in \{\mathcal{N}_{s,m}\}} \max_{t \in \mathcal{N}_{s,m}} [a_s(l-1) - a_t(l-1)]^2 \\ &\quad + \frac{1}{2N} \sum_{s=1}^N (1-p)^{|\mathcal{N}_s|} \frac{1}{|\mathcal{N}_s|} \sum_{t \in \mathcal{N}_s} [a_s(l-1) - a_t(l-1)]^2 \\ &\quad - \frac{1}{2N} \sum_{s=1}^N \max_{t \in \mathcal{N}_s} [a_s(l-1) - a_t(l-1)]^2 \end{aligned} \quad (3.41)$$

Substituting Eq. (3.41) into Eqs. (3.39) and (3.40) gives the convergence bound of GG as

$$\begin{aligned} & \mathbb{E} \left[\left\| \mathbf{W}^{GG}(1:l)\mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right] \\ & \leq [\lambda_2(\bar{\mathbf{W}}) - \beta_l - \xi_l] \mathbb{E} \left[\left\| \mathbf{W}^{GG}(1:l-1)\mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right] \end{aligned} \quad (3.42)$$

where

$$\begin{aligned} \gamma_l &= \frac{1}{2N} \sum_{s=1}^N \max_{t \in \mathcal{N}_s} [a_s(l-1) - a_t(l-1)]^2 \\ & - \frac{1}{2N} \sum_{s=1}^N \sum_{m=1}^{|\mathcal{N}_s|} p^m (1-p)^{|\mathcal{N}_s|-m} \sum_{\mathcal{N}_{s,m} \in \{\mathcal{N}_{s,m}\}} \max_{t \in \mathcal{N}_{s,m}} [a_s(l-1) - a_t(l-1)]^2 \\ & - \frac{1}{2N} \sum_{s=1}^N (1-p)^{|\mathcal{N}_s|} \frac{1}{|\mathcal{N}_s|} \sum_{t \in \mathcal{N}_s} [a_s(l-1) - a_t(l-1)]^2 \end{aligned} \quad (3.43)$$

$$\xi_l = \frac{\gamma_l}{\mathbb{E} \left[\left\| \mathbf{W}^{GG}(1:l-1)\mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right]} \quad (3.44)$$

It follows from Lemma 3.1 that $\xi_l \geq 0$ and the equality holds if and only if $\mathbf{a}(l-1) = \bar{\mathbf{a}}$. Repeatedly using Eq. (3.42) gives

$$\begin{aligned} \mathbb{E} \left[\left\| \mathbf{W}^{GG}(1:l)\mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right] & \leq \left\| \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \prod_{i=1}^l (\lambda_2(\bar{\mathbf{W}}) - \beta_i - \xi_i) \\ & \leq \left\| \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \left(\lambda_2(\bar{\mathbf{W}}) - \min_{i \in [l]} \{\beta_i\} - \min_{i \in [l]} \{\xi_i\} \right)^l \end{aligned} \quad (3.45)$$

Using Eq. (3.45) with the help of Markov's inequality, we have

$$\begin{aligned} \Pr \left(\frac{\left\| \mathbf{W}^{GG}(1:l)\mathbf{a}(0) - \bar{\mathbf{a}} \right\|}{\left\| \mathbf{a}(0) - \bar{\mathbf{a}} \right\|} \geq \epsilon \right) &= \Pr \left(\frac{\left\| \mathbf{W}^{GG}(1:l)\mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2}{\left\| \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2} \geq \epsilon^2 \right) \\ &\leq \frac{\mathbb{E} \left[\left\| \mathbf{W}^{GG}(1:l)\mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2 \right]}{\epsilon^2 \left\| \mathbf{a}(0) - \bar{\mathbf{a}} \right\|^2} \\ &\leq \epsilon^{-2} \left(\lambda_2(\bar{\mathbf{W}}) - \min_{i \in [l]} \{\beta_i\} - \min_{i \in [l]} \{\xi_i\} \right)^l \end{aligned} \quad (3.46)$$

which means that the ϵ -averaging time for GG is upper bounded by

$$T_{ave}^{GG}(\epsilon) \leq \frac{3 \log \epsilon^{-1}}{\log \left(\lambda_2(\bar{\mathbf{W}}) - \min_{i \in [l]} \{\beta_i\} - \min_{i \in [l]} \{\xi_i\} \right)^{-1}} \quad (3.47)$$

Recall that the upper bound for the ϵ -averaging time of RG is given by [36]

$$T_{ave}^{RG}(\epsilon) \leq \frac{3 \log \epsilon^{-1}}{\log \lambda_2(\overline{\mathbf{W}})^{-1}} \quad (3.48)$$

Combining Eqs. (3.37), (3.47) and (3.48), it is easy to verify Eq. (3.38), which completes the proof. $\square$

Remark 3.5. *Note that Theorem 3.1 only provides theoretical comparison of the upper bound of ϵ -averaging time for different gossip algorithms. However, extensive numerical analysis shows that the proposed algorithm achieves very comparable performance to that of the GG with significantly reduced communication burden.*

3.4 Distributed Kalman Filter Design and Analysis

This section first develops a DKF using the proposed SGG algorithm based on a synchronous time model. The boundedness of the estimation error covariance matrix with finite number of gossip iterations will then be presented.

3.4.1 Filter Development

This chapter considers a linear stochastic discrete-time system with N sensors as:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{F}_k \mathbf{x}_k + \mathbf{w}_k \\ \mathbf{z}_{k,i} &= \mathbf{H}_{k,i} \mathbf{x}_k + \mathbf{v}_{k,i}, \quad i = 1, 2, \dots, N \end{aligned} \quad (3.49)$$

where $\mathbf{x}_k \in \mathbb{R}^n$ denotes the system state vector at time instant k ; $\mathbf{z}_{k,i} \in \mathbb{R}^{m_i}$ represents the measurement vector of the i th sensor at time instant k . Matrices $\mathbf{F}_k \in \mathbb{R}^{n \times n}$ and $\mathbf{H}_{k,i} \in \mathbb{R}^{m_i \times n}$ stand for the system transition and measurement matrices, respectively. $\mathbf{w}_k \in \mathbb{R}^n$ and $\mathbf{v}_{k,i} \in \mathbb{R}^{m_i}$ are uncorrelated white noises with zero mean and variances $\mathbf{Q}_k \in \mathbb{R}^{n \times n}$ and $\mathbf{R}_{k,i} \in \mathbb{R}^{m_i \times m_i}$. For simplicity, it is usually assumed that the measurement noise is uncorrelated across the sensor nodes.

At time instant k , every local sensor node predict the target state based on previous

estimate and available mode information as

$$\begin{aligned}\mathbf{x}_{k|k-1,i} &= \mathbf{F}_k \mathbf{x}_{k-1|k-1,i} \\ \mathbf{P}_{k|k-1,i} &= \mathbf{F}_{k-1} \mathbf{P}_{k-1|k-1,i} \mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}\end{aligned}\tag{3.50}$$

where $\mathbf{x}_{k|k-1,i}$ and $\mathbf{x}_{k|k,i}$, respectively, represent the state prediction and update at time instant k from sensor i with corresponding error covariances $\mathbf{P}_{k|k-1,i}$ and $\mathbf{P}_{k|k,i}$.

After obtaining the one-step prediction by local Kalman filter, we are now interested in updating the local prediction via fusing information from locally-connected neighbours. As the information weighted fusion rule or the so-called parallel fusion has advantages of preserving local estimate consistency and guaranteed global convergence to the centralised solution [32, 106], this fusion rule is leveraged in this chapter to fuse the local estimates. That is, the information to be shared between two local sensor nodes is defined as

$$\begin{aligned}\mathbf{u}_{k,i} &= \frac{1}{N} \mathbf{P}_{k|k-1,i}^{-1} \mathbf{x}_{k|k-1,i} + \mathbf{H}_{k,i}^T \mathbf{R}_{k,i}^{-1} \mathbf{z}_{k,i} \\ \mathbf{U}_{k,i} &= \frac{1}{N} \mathbf{P}_{k|k-1,i}^{-1} + \mathbf{H}_{k,i}^T \mathbf{R}_{k,i}^{-1} \mathbf{H}_{k,i}\end{aligned}\tag{3.51}$$

Note that the key of implementing SGG in sensor fusion is to define a proper criterion to evaluate the similarity between two local estimates. For this reason, a statistical distance to quantify the similarity between $(\mathbf{u}_{k,i}, \mathbf{U}_{k,i})$ and $(\mathbf{u}_{k,j}, \mathbf{U}_{k,j})$ will be defined first. Notice that the residual of $\mathbf{u}_{k,i}$ is give by

$$\begin{aligned}\tilde{\mathbf{u}}_{k,i} &= \frac{1}{N} \mathbf{P}_{k|k-1,i}^{-1} (\mathbf{x}_{k|k-1,i} - \mathbf{x}_k) + \mathbf{H}_{k,i}^T \mathbf{R}_{k,i}^{-1} (\mathbf{z}_{k,i} - \mathbf{H}_{k,i} \mathbf{x}_k) \\ &= \frac{1}{N} \mathbf{P}_{k|k-1,i}^{-1} (\mathbf{x}_{k|k-1,i} - \mathbf{x}_k) + \mathbf{H}_{k,i}^T \mathbf{R}_{k,i}^{-1} \mathbf{v}_{k,i}\end{aligned}\tag{3.52}$$

Assume that $\mathbf{x}_{k|k-1,i}$ is unbiased, i.e., $\mathbb{E} [\mathbf{x}_{k|k-1,i}] = \mathbf{x}_k$. Then, it is straightforward to verify that the covariance of $\tilde{\mathbf{u}}_{k,i}$ is determined as

$$\mathbb{E} [\tilde{\mathbf{u}}_{k,i} \tilde{\mathbf{u}}_{k,i}^T] = \mathbf{U}_{k,i}\tag{3.53}$$

Based on Eq. (3.53), the statistical difference between $(\tilde{\mathbf{u}}_{k,i}, \mathbf{U}_{k,i})$ and $(\tilde{\mathbf{u}}_{k,j}, \mathbf{U}_{k,j})$ can then be quantified by the well-known Mahalanobis distance

$$d_{i,j} = (\tilde{\mathbf{u}}_{k,i} - \tilde{\mathbf{u}}_{k,j})^T (\mathbf{U}_{k,i} + \mathbf{U}_{k,j})^{-1} (\tilde{\mathbf{u}}_{k,i} - \tilde{\mathbf{u}}_{k,j})\tag{3.54}$$

The issue of utilising the Mahalanobis distance $d_{i,j}$ as the similarity measure is that

the information on true target state $\mathbf{x}_k$ is not available in practice. To address this problem, we assume that the previous time instant achieves fully average consensus, i.e., $\mathbf{x}_{k-1|k-1,i} = \mathbf{x}_{k-1|k-1,j}$ and $\mathbf{P}_{k-1|k-1,i} = \mathbf{P}_{k-1|k-1,j}$, $\forall i \neq j$. Note that this is a typical assumption in developing DKFs for sensor networks [32]. With this in mind, Eq. (3.54) reduces to

$$d_{i,j} = (\mathbf{u}_{k,i} - \mathbf{u}_{k,j})^T (\mathbf{U}_{k,i} + \mathbf{U}_{k,j})^{-1} (\mathbf{u}_{k,i} - \mathbf{u}_{k,j}) \quad (3.55)$$

By exploiting the Mahalanobis distance $d_{i,j}$ as a similarity measure between two local estimates, we can then implement the proposed SGG algorithm to find a suboptimal communication path for every local sensor node to fuse $(\mathbf{u}_{k,i}, \mathbf{U}_{k,i})$. After L iterations of the gossip process, the measurement update of the proposed DKF is then given by

$$\begin{aligned} \mathbf{x}_{k|k,i} &= [\mathbf{U}_{k,i}(L)]^{-1} \mathbf{u}_{k,i}(L) \\ \mathbf{P}_{k|k,i}^{-1} &= N \mathbf{U}_{k,i}(L) \end{aligned} \quad (3.56)$$

The complete pseudo code of the proposed SGG-DKF is summarised in Algorithm 3.2.

Remark 3.6. Define $\mathbf{y}_{k,i} = \mathbf{P}_{k|k-1,i}^{-1} \mathbf{x}_{k|k-1,i}$, $\mathbf{Y}_{k,i} = \mathbf{P}_{k|k-1,i}^{-1}$, $\mathbf{q}_{k,i} = \mathbf{H}_{k,i}^T \mathbf{R}_{k,i}^{-1} \mathbf{z}_{k,i}$ and $\mathbf{\Omega}_{k,i} = \mathbf{H}_{k,i}^T \mathbf{R}_{k,i}^{-1} \mathbf{H}_{k,i}$. As the proposed SGG algorithm developed guarantees asymptotic convergence, it is straightforward to verify that

$$\begin{aligned} \lim_{L \rightarrow \infty} \mathbf{u}_{k,i}(L) &= \frac{1}{N} \sum_{i=1}^N \frac{\mathbf{y}_{k,i}(0)}{N} + \frac{1}{N} \sum_{i=1}^N \mathbf{q}_{k,i}(0) \\ \lim_{L \rightarrow \infty} \mathbf{U}_{k,i}(L) &= \frac{1}{N} \sum_{i=1}^N \frac{\mathbf{Y}_{k,i}(0)}{N} + \frac{1}{N} \sum_{i=1}^N \mathbf{\Omega}_{k,i}(0) \end{aligned} \quad (3.57)$$

As gossip on priors guarantees that all local sensors have the same priori estimates, e.g., $\mathbf{y}_{k,i} = \frac{1}{N} \sum_{i=1}^N \mathbf{y}_{k,i}(0)$ and $\mathbf{Y}_{k,i} = \frac{1}{N} \sum_{i=1}^N \mathbf{Y}_{k,i}(0)$, it is immediate to see that the proposed DKF with infinite number of iterations can recover the performance of centralised estimation by substituting Eq. (3.57) into the measurement update step in Algorithm 3.2.

Remark 3.7. Compared to previously-developed gossip-based DKFs [39, 40], the following two improvements are made in the proposed approach: (1) a new SSG algorithm is utilised to exploit the benefits of both RG and GG; and (2) the information weighted fusion rule is leveraged in the proposed algorithm while [39, 40] utilised the concept of MVF.

Remark 3.8. Note that the proposed SGG-DKF is developed based on assumption that the previous local estimates are converged. In real applications, however, only finite number of gossip iterations are acceptable, which means that $\mathbf{x}_{k-1|k-1,i} \neq \mathbf{x}_{k-1|k-1,j}$ and $\mathbf{P}_{k-1|k-1,i} \neq \mathbf{P}_{k-1|k-1,j}$. Therefore, all local estimates are auto-correlated during the fusion phase and thus SGG-DKF also suffers from the well-known auto-correlation problem. However, as the information weighted consensus concept incorporates CI, which is proved to be highly robust against the auto-correlation among local estimates. This issue, therefore, can also be alleviated by the proposed SGG-DKF.

3.4.2 Boundedness Analysis

Although the proposed SGG-DKF algorithm is shown to be asymptotically optimal, only limited number of gossip iterations is available in real applications. For this reason, this subsection analyses the boundedness of the proposed algorithm with finite number of gossip iterations. Before giving the main results, the following two general assumptions are made.

Assumption 3.1. The system matrix $\mathbf{F}_k$ is invertible.

Assumption 3.2. The sensor network is collectively observable, e.g., the pair $(\mathbf{F}_k, \mathbf{H}_k)$ is observable, where $\mathbf{H}_k = [\mathbf{H}_{k,1}^T, \mathbf{H}_{k,2}^T, \dots, \mathbf{H}_{k,N}^T]^T$.

Note that not all sensor nodes can detect the target in reality due to limited sensing capability. This means that it is unreasonable to assume the local observability, e.g., the pair $(\mathbf{F}_k, \mathbf{H}_{k,i})$ is observable. For this reason, the collective observability is utilised to define the observability of the sensor network. It is clear that the network is collectively observable if and only if the network is connected, e.g., any two sensors are direct (one-hop) or indirect connected (multi-hop). The result of the boundedness of the fused covariance matrices is now presented in the following theorem.

Theorem 3.5. Assume that $\mathbf{P}_{k-1|k-1,i}$ is positive-definite and under Assumptions 3.1 and 3.2, there exists an time instant b and a positive scalar α such that the fused covariance is uniformly bounded in average sense as

$$\mathbb{E} [\mathbf{P}_{k+b|k+b,i}] \leq \alpha \mathbf{I}_n \quad (3.58)$$

where the expectation is over all possible network connections.

Algorithm 3.2 Sample greedy gossip distributed Kalman filter at the i th node

Input: Previous target estimation $\{\mathbf{x}_{k-1|k-1,i}, \mathbf{P}_{k-1|k-1,i}\}$, received measurements

 $\mathbf{z}_{k,i}$
Output: Current estimation $\{\mathbf{x}_{k|k,i}, \mathbf{P}_{k|k,i}\}$

(1) Prediction:

$$\begin{aligned}\mathbf{x}_{k|k-1,i} &= \mathbf{F}_k \mathbf{x}_{k-1|k-1,i} \\ \mathbf{P}_{k|k-1,i} &= \mathbf{F}_{k-1} \mathbf{P}_{k-1|k-1,i} \mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}\end{aligned}$$

(2) Compute the information terms:

$$\begin{aligned}\mathbf{u}_{k,i} &= \frac{1}{N} \mathbf{P}_{k|k-1,i}^{-1} \mathbf{x}_{k|k-1,i} + \mathbf{H}_{k,i}^T \mathbf{R}_{k,i}^{-1} \mathbf{z}_{k,i} \\ \mathbf{U}_{k,i} &= \frac{1}{N} \mathbf{P}_{k|k-1,i}^{-1} + \mathbf{H}_{k,i}^T \mathbf{R}_{k,i}^{-1} \mathbf{H}_{k,i}\end{aligned}$$

(3) **for** $l = 0, 1, \dots, L$ **do**

 Compute the Mahalanobis distance $d_{i,j}$

 Use Algorithm algo:sgg1 to find a local neighbour j^* which has the largest value of $d_{i,j^*}(l)$ among the active sensor nodes as

$$j^* = \begin{cases} \max_{j \in \mathcal{A}_i} d_{i,j}(l), & \mathcal{A}_i \neq \emptyset \\ \text{randomly picking from } \mathcal{N}_i, & \mathcal{A}_i = \emptyset \end{cases}$$

 After finding j^* , perform information averaging

$$\begin{aligned}\mathbf{u}_{k,i}(l) &= \frac{1}{2} [\mathbf{u}_{k,i}(l-1) + \mathbf{u}_{k,j^*}(l-1)] \\ \mathbf{U}_{k,i}(l) &= \frac{1}{2} [\mathbf{U}_{k,i}(l-1) + \mathbf{U}_{k,j^*}(l-1)]\end{aligned}$$

(4) Measurement update:

$$\begin{aligned}\mathbf{x}_{k|k,i} &= [\mathbf{U}_{k,i}(L)]^{-1} \mathbf{u}_{k,i}(L) \\ \mathbf{P}_{k|k,i}^{-1} &= N \mathbf{U}_{k,i}(L)\end{aligned}$$

Proof. According to the propagation of the prediction step in Kalman filter, we have

$$\begin{aligned}\mathbf{P}_{k|k-1,i}^{-1} &= (\mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1,i}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1})^{-1} \\ &= (\mathbf{F}_{k-1}^{-1})^T \left[\mathbf{P}_{k-1|k-1,i} + \mathbf{F}_{k-1}^{-1} \mathbf{Q}_{k-1} (\mathbf{F}_{k-1}^{-1})^T \right]^{-1} \mathbf{F}_{k-1}^{-1}\end{aligned}\quad (3.59)$$

Since the covariance of the process noise is bounded, there exists a positive scalar $\beta > 0$ such that

$$\mathbf{F}_{k-1}^{-1} \mathbf{Q}_{k-1} (\mathbf{F}_{k-1}^{-1})^T \leq \beta \mathbf{P}_{k-1|k-1,i} \quad (3.60)$$

By choosing $\gamma = (1 + \beta)^{-1}$, Eq. (3.59) can be reformulated as

$$\mathbf{P}_{k|k-1,i}^{-1} \geq \gamma (\mathbf{F}_{k-1}^{-1})^T \mathbf{P}_{k-1|k-1,i}^{-1} \mathbf{F}_{k-1}^{-1} \quad (3.61)$$

After performing L steps of gossip iterations, we have

$$\mathbf{P}_{k|k,i}^{-1} = \sum_{j=1}^N w_{i,j}^L \mathbf{P}_{k|k-1,j}^{-1} + N \sum_{j=1}^N w_{i,j}^L \mathbf{H}_{k,j}^T \mathbf{R}_{k,j}^{-1} \mathbf{H}_{k,j} \quad (3.62)$$

where $w_{i,j}^L$ denotes the (i, j) th element of matrix $\mathbf{W}^{SGG}(1 : L)$.

Substituting Eq. (3.61) into Eq. (3.62) gives

$$\mathbf{P}_{k|k,i}^{-1} \geq \gamma \sum_{j=1}^N w_{i,j}^L (\mathbf{F}_{k-1}^{-1})^T \mathbf{P}_{k-1|k-1,j}^{-1} \mathbf{F}_{k-1}^{-1} + N \sum_{j=1}^N w_{i,j}^L \mathbf{H}_{k,j}^T \mathbf{R}_{k,j}^{-1} \mathbf{H}_{k,j} \quad (3.63)$$

Applying the preceding inequality to $\mathbf{P}_{k+b|k+b,i}^{-1}$ for b steps repeatedly yields

$$\begin{aligned}\mathbf{P}_{k+b|k+b,i}^{-1} &\geq \gamma^b \sum_{j=1}^N w_{i,j}^{bL} (\mathbf{F}_{k-1}^{-b})^T \mathbf{P}_{k-1|k-1,j}^{-1} \mathbf{F}_{k-1}^{-b} \\ &\quad + N \sum_{j=1}^N \sum_{m=1}^b \gamma^{b-m} w_{i,j}^{(b-m+1)L} (\mathbf{F}_{k-1}^{m-b})^T \mathbf{H}_{k,j}^T \mathbf{R}_{k,j}^{-1} \mathbf{H}_{k,j} \mathbf{F}_{k-1}^{m-b} \\ &\geq N \sum_{j=1}^N \sum_{m=1}^b \gamma^{b-m} w_{i,j}^{(b-m+1)L} (\mathbf{F}_{k-1}^{m-b})^T \mathbf{H}_{k,j}^T \mathbf{R}_{k,j}^{-1} \mathbf{H}_{k,j} \mathbf{F}_{k-1}^{m-b}\end{aligned}\quad (3.64)$$

Denote the diameter of the undirected graph $\mathcal{G}$ as $\mathcal{D}(\mathcal{G})$. Then, there exists a path no longer than $\mathcal{D}(\mathcal{G})$ that any two nodes are connected for a connected network. Also note that the proposed SGG process ensures the possibility that each local sensor node can communicate with all its connected neighbours. This means that if

the number of gossip iterations satisfies $L \geq \mathcal{D}(\mathcal{G})$, we have

$$\mathbb{E} \left[w_{i,j}^{(b-m+1)L} \right] > 0 \quad (3.65)$$

Therefore, it can be concluded that there exist positive constants $\pi_{i,j}$ such that

$$\mathbb{E} \left[w_{i,j}^{(b-m+1)L} \right] \geq \pi_{i,j} > 0 \quad (3.66)$$

Substituting Eq. (3.66) into Eq. (3.64) gives

$$\begin{aligned} \mathbb{E} \left[\mathbf{P}_{k+b|k+b,i}^{-1} \right] &\geq \sum_{j=1}^N \sum_{m=1}^b \gamma^{b-m} \pi_{i,j} (\mathbf{F}_{k-1}^{m-b})^T \mathbf{H}_{k,j}^T \mathbf{R}_{k,j}^{-1} \mathbf{H}_{k,j} \mathbf{F}_{k-1}^{m-b} \\ &\geq \gamma^{b-1} \pi_{i,\min} \sum_{j=1}^N \sum_{m=1}^b (\mathbf{F}_{k-1}^{m-b})^T \mathbf{H}_{k,j}^T \mathbf{R}_{k,j}^{-1} \mathbf{H}_{k,j} \mathbf{F}_{k-1}^{m-b} \end{aligned} \quad (3.67)$$

where $\pi_{i,\min} = \min_j \{\pi_{i,j}\}$. Since the sensor network is collectively observable, there exists $b > 0$ such that the Grammian matrix satisfies $\sum_{m=1}^b (\mathbf{F}_{k-1}^{m-b})^T \mathbf{H}_{k,j}^T \mathbf{R}_{k,j}^{-1} \mathbf{H}_{k,j} \mathbf{F}_{k-1}^{m-b} > 0$ for all j . Additionally, noting that the covariance matrix of the measurement noise is always positive-definite and bounded, we have

$$\sum_{j=1}^N \sum_{m=1}^b (\mathbf{F}_{k-1}^{m-b})^T \mathbf{H}_{k,j}^T \mathbf{R}_{k,j}^{-1} \mathbf{H}_{k,j} \mathbf{F}_{k-1}^{m-b} \geq \sigma I_n > 0 \quad (3.68)$$

where $\sigma > 0$. This subsequently indicates that there exists a positive scalar α such that the fused covariance is uniformly bounded in the average sense as $\mathbb{E} [\mathbf{P}_{k+b|k+b,i}] \leq \alpha I_n$ where $\alpha = (\gamma^{b-1} \pi_{i,\min} \sigma)^{-1}$. $\square$

3.5 Numerical Simulations

This section presents a performance evaluation of the proposed SGG-DKF algorithm with comparisons to other approaches using Monte-Carlo simulations.

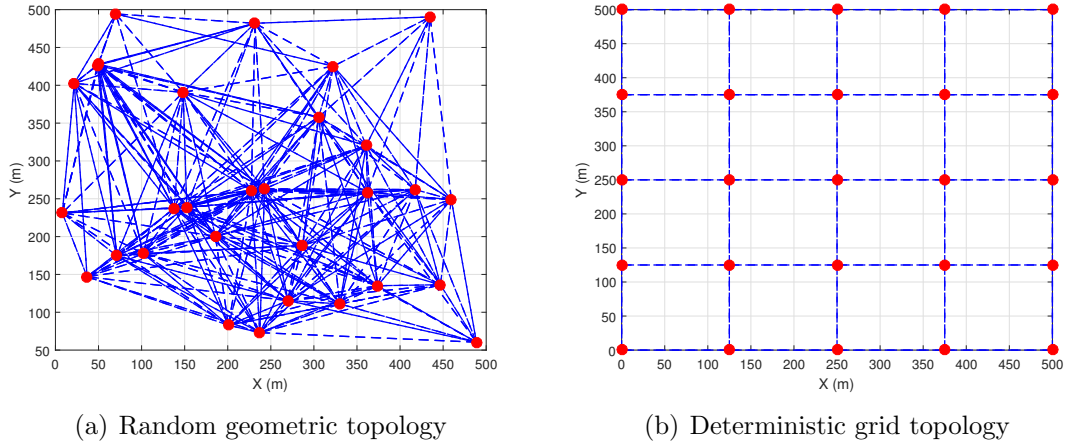


Figure 3.1: Examples of two different network topologies. The red circles denote the sensor locations and the blue lines refer to the connections between sensor nodes. (a) with 30 sensors being randomly placed inside the surveillance region and two sensors being connected if their relative distance is less than $300m$. (b) with 25 sensors being placed as a grid shape inside the surveillance region.

3.5.1 Simulation Setup

The target in the considered scenario randomly moves in a $500m \times 500m$ rectangular surveillance area. For performance evaluation, we consider two different types of network topologies: random geometric network with 30 sensors, and deterministic grid network with 25 sensors. Note that these two types of topologies are widely utilised in analysing the performance of distributed network-wide computation algorithms [103]. For the random geometric network, each sensor is randomly placed inside the surveillance region. Two sensors are connected if their relative distance is less than $300m$ in our simulations. Examples of these two different sensor topologies are presented Fig. 3.1.

Each target's state is represented by a 4-D vector, with 2-D position and 2-D velocity components. In estimation update, the system equation is assumed to be the well-known constant velocity (CV) model, e.g.,

$$\mathbf{F}_k = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.69)$$

with $T_s = 1s$ being the sampling time. The variance of process noise of the consid-

ered constant velocity model is determined as

$$\mathbf{Q}_k = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.70)$$

Each sensor collects position measurements at regular time instants $t_k = kT_s$, $k \in \{1, 2, \dots, 100\}$, as

$$\mathbf{H}_{k,i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (3.71)$$

The measurement noise is subject to a Gaussian white noise as $v_{k,i} \sim \mathcal{N}(\cdot; 0, \mathbf{R}_{k,i})$ with $\mathbf{R}_{k,i} = \text{diag}(\sigma_r^2, \sigma_r^2)$, $\sigma_r = 10m$. For initialisation, the covariance matrix of the target at sensor node i is chosen as $\mathbf{P}_{0|0,i} = \text{diag}(100, 100, 10, 10)$. The initial state estimates are generated from a Gaussian distribution around the true target state with the covariance $\mathbf{P}_{0|0,i}$. The starting point of the target is also randomly generated inside the surveillance region at every Monte-Carlo run.

3.5.2 Performance Metric

Let $\mathbf{x}_{k|k,i}^j$ denote the estimated state of the target at sensor node i at time instant k of the j th Monte Carlo run and $\mathbf{x}_k^j$ represent the true target state at time instant k of the j th Monte Carlo run. The mean error (ME) of position estimation and root-mean-square error (RMSE) of position estimation at time instant k , averaged over M Monte-Carlo runs and N sensors, is defined as

$$\begin{aligned} \text{ME}_k^{\text{pos}} &= \frac{1}{MN} \sum_{i=1}^N \sum_{j=1}^M \left\| \mathbf{p}_{k|k,i}^j - \mathbf{p}_k^j \right\| \\ \text{RMSE}_k^{\text{pos}} &= \left(\frac{1}{MN} \sum_{i=1}^N \sum_{j=1}^M \left\| \mathbf{p}_{k|k,i}^j - \mathbf{p}_k^j \right\|^2 \right)^{\frac{1}{2}} \end{aligned} \quad (3.72)$$

where $\mathbf{p}_k^j = \mathbf{x}_k^j(1:2)$ and $\mathbf{p}_{k|k,i}^j = \mathbf{x}_{k|k,i}^j(1:2)$ are true and estimated positions of the j th target.

For performance evaluation of the proposed algorithm, the time averaged ME and

RMSE are utilised. These two metrics are computed as

$$\begin{aligned} \text{ME}_{\text{avg}}^{\text{pos}} &= \frac{1}{T} \sum_{k=1}^T \text{ME}_k^{\text{pos}} \\ \text{RMSE}_{\text{avg}}^{\text{pos}} &= \frac{1}{T} \sum_{k=1}^T \text{RMSE}_k^{\text{pos}} \end{aligned} \tag{3.73}$$

where $T = 100$ is the total number of time instants during the tracking period.

3.5.3 Comparison with Different Gossip Algorithms

This subsection investigates performance of the proposed SGG-DKF algorithm, compared with randomised gossip DKF (RG-DKF) and greedy gossip DKF (GG-DKF). The main objective of the performance comparison in this subsection is to validate the theoretical analysis results of the proposed SGG algorithm. For fair comparison, RG-DKF and GG-DKF algorithms are obtained by replacing the SGG process with RG and GG in SGG-DKF. The detailed pseudo codes of these two algorithms are presented in Appendix A. Note that these two DKFs are also new algorithms that have never been proposed in the existing literatures. To better demonstrate the characteristics of different gossip algorithms, it is assumed that each sensor node has unlimited sensing range in this subsection.

In gossip-based distributed estimation, information transmission via multiple rounds of communication among locally-connected sensors are required and the performance highly depends on the number of iterations, i.e., L . In order to investigate the effect of the parameter L on the fusion performance, Monte-Carlo comparisons of different gossip-based DKFs are carried out with respect to different number of iterations $L = 1, 2, \dots, 10$. In the simulations, the sensor activation probability for implementing SGG is set as $p = 0.5$. The simulation results of ME of target position estimation and RMSE of target position estimation obtained from 500 Monte-Carlo runs are depicted in Fig. 3.2. As shown in the figure, it can be noted that all tested DKFs asymptotically converge to the Bayesian optimal CKF. From Fig. 3.2, it can be noted that RG-DKF has the lowest convergence speed among these three different gossip-based DKFs. As GG-DKF picks up the optimal communication path for every local sensor node at each gossip iteration, it exhibits the fastest convergence rate at the expense of high communication burden. The proposed SGG-DKF only leverages a suboptimal communication path, e.g., performing greedy

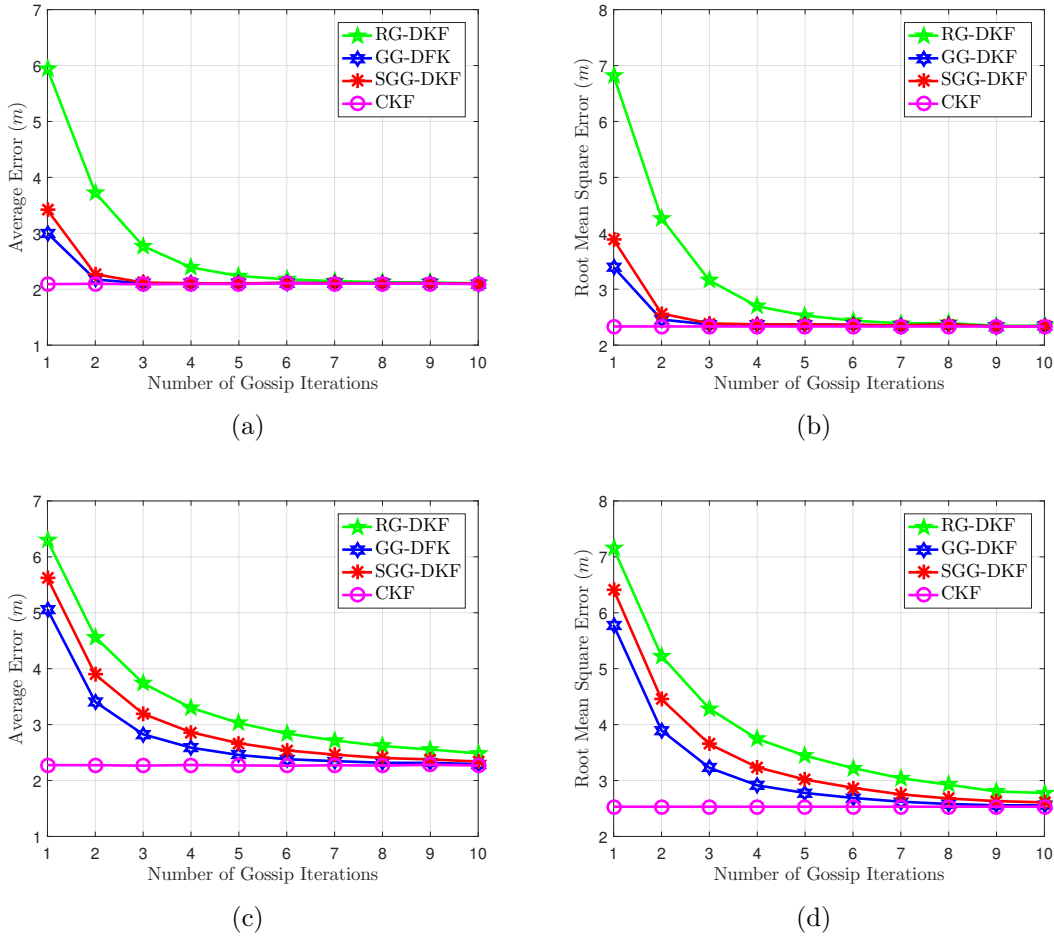


Figure 3.2: Monte-Carlo comparison results with respect to different number of gossip iterations. (a) ME comparison results for random geometric sensor network. (b) RMSE comparison results for random geometric sensor network. (c) ME comparison results for deterministic grid sensor network. (d) RMSE comparison results for deterministic grid sensor network.

sensor selection within a set of randomly-chosen active sensor nodes. Therefore, the SGG-DKF provides tradeoff convergence performance between RG-DKF and GG-DKF. As the probability threshold in selecting the active node is $p = 0.5$, the proposed algorithm only requires half communication burden in the average sense at each iteration, compared to the GG-DKF. These results conform with the analytic findings presented in Sec. 3.3. Interestingly, the performance of SGG-DKF is very comparable to that of the GG-DKF and its convergence rate is much faster than that of the RG-DKF for the random geometric sensor network topology even with $p = 0.5$.

Now, let us investigate the effect of the sensor activation probability on the tracking performance. For this purpose, the number of gossip iterations in each Monte-Carlo

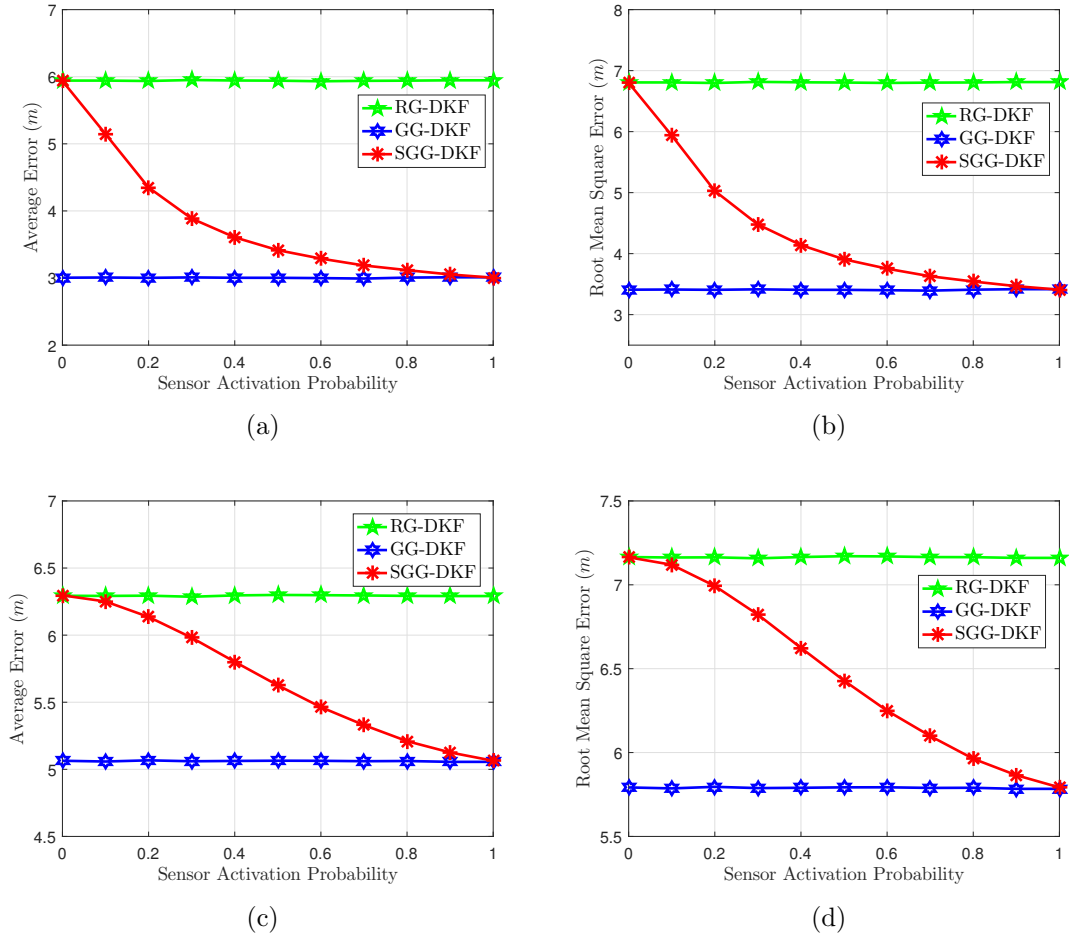


Figure 3.3: Monte-Carlo comparison results with respect to different sensor activation probabilities. (a) ME comparison results for random geometric sensor network. (b) RMSE comparison results for random geometric sensor network. (c) ME comparison results for deterministic grid sensor network. (d) RMSE comparison results for deterministic grid sensor network.

run is set as $L = 1$. Fig. 3.3 presents the comparison results of target position estimation ME and RMSE for different gossip-based DKFs with different sensor activation probabilities $p = 0, 0.1, \dots, 1$ obtained from 500 Monte-Carlo runs. From this figure, it can be observed that GG-DKF provides the best estimation performance among all the tested algorithms. This can be attributed to the fact that GG-DKF finds the optimal local sensor node for fusion. However, as stated before, this achievement requires each local sensor to communicate with all its connected neighbours at each iteration. As a comparison, the proposed SGG-DKF offers great flexibility and well balance between communication cost and convergence performance introduced by the stochastic sampling strategy. With the increasing of sensor activation probability, SGG-DKF provides improved estimation performance and converges to that of GG-DKF when $p = 1$. If the local sensor node cannot provide

enough bandwidth for communication, a relatively small sensor activation probability can be selected to save the communication cost. When $p = 0$, the proposed algorithm becomes identical to RG-DKF. The results confirm that the proposed SGG algorithm is a generalised version of the randomised and greedy gossip algorithms.

3.5.4 Comparison with Previous Gossip-Based Distributed Estimators

This subsection evaluates the advantage of the proposed SGG-DKF when the sensing ability is limited, which is the case in practice. For rigorous evaluation, we perform Monte-Carlo comparisons with previous gossip-based DKFs, i.e., RG-MVF [39] and GG-MVF [40]. Each sensor node is assumed to be able to detect the target if the relative distance between the target and the sensor node is less than $100m$. In all simulations, the sensor activation probability for implementing SGG-DKF is set as $p = 0.5$.

The simulation results of target position estimation ME and RMSE obtained from 500 Monte-Carlo runs are depicted in Fig. 3.4. It follows from Fig. 3.4 that the SGG-DKF still significantly outperforms both RG-MVF and GG-MVF, especially for the grid sensor network. This can be attributed to the fact that the proposed SGG-DKF utilises the information weighted fusion concept while RG-MVF and GG-MVF only leverage MVF in the fusion process. If one sensor cannot detect the target due to limited sensing range, the sensor node can only use target state prediction or the so-called prior estimation as the local state estimation, which generally has certain amount of estimation errors. Since RG-MVF and GG-MVF never utilise local prior knowledge in the fusion process, these two algorithms constrain the posterior estimates as the prior estimates if the sensor and its neighbours cannot detect the target due to limited gossip iterations. As a comparison, the prior estimation of each sensor node is weighted by the inverse of the corresponding error covariance in the proposed SGG-DKF during fusion. This handles the issue of naive sensors, e.g., target is outside of the sensor's field-of-view, by placing less weight when receiving the information from a naive neighbour sensor. Therefore, the proposed SGG-DKF is helpful in ensuring the consistency of local estimates and is demonstrated to generate better tracking performance when the number of gossip iterations is small.

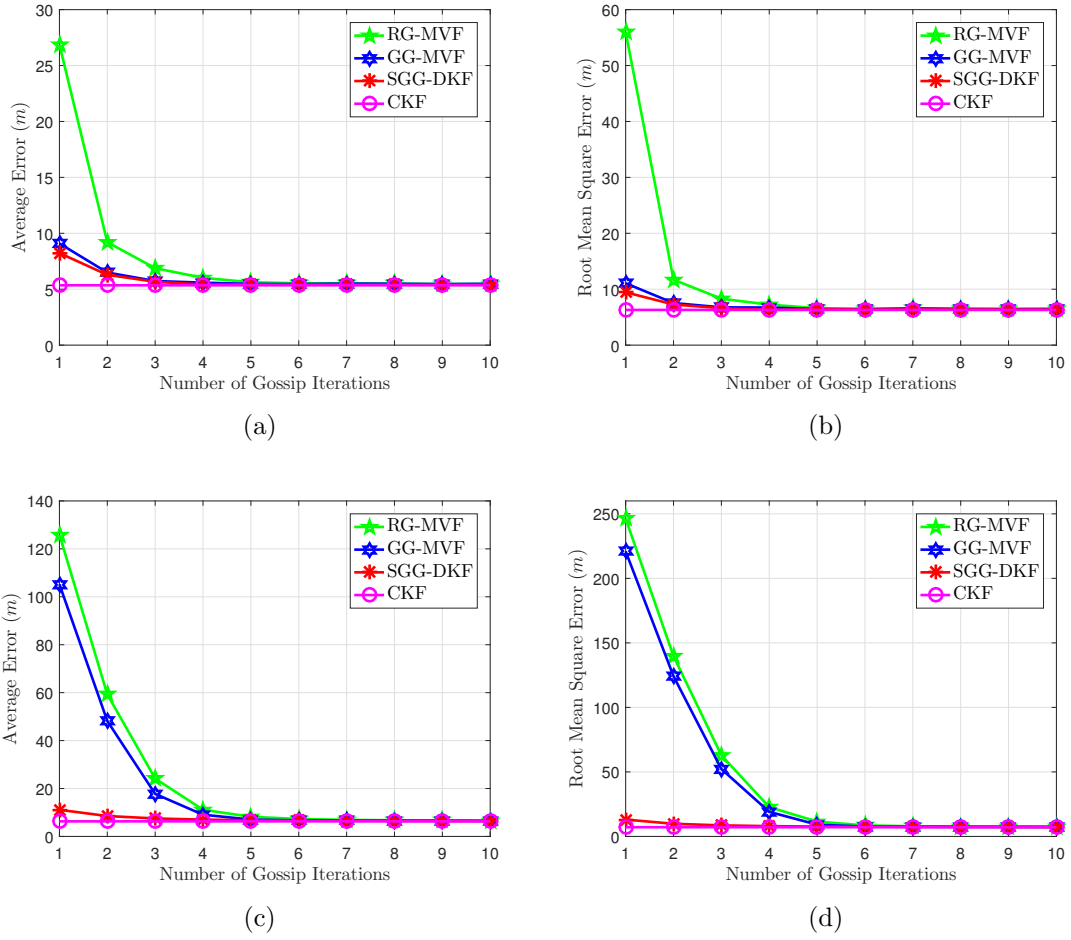


Figure 3.4: Monte-Carlo comparison results of different gossip DKFs. (a) ME comparison results for random geometric sensor network. (b) RMSE comparison results for random geometric sensor network. (c) ME comparison results for deterministic grid sensor network. (d) RMSE comparison results for deterministic grid sensor network.

3.6 Conclusions

This chapter proposes a distributed SGG-DKF for target tracking over a sensor network. The proposed algorithm utilises the concept of information weighted fusion in conjunction with a new SGG process. Rigorous convergence analysis of the proposed SGG algorithm is carried out to support its applications. Theoretical analysis reveals that the proposed DKF guarantees asymptotic convergence to the optimal CKF. The empirical investigation demonstrates the validity of the theoretical analysis results. Also, theoretical and numerical analyses indicate that the proposed algorithm can be viewed as a generalised version of the RG and GG algorithms. The prominent feature of the proposed algorithm lies in that it allows tradeoff between convergence

speed and communication burden: our algorithm shows faster convergence speed, compared to the RG-DKF and requires less communication burden, compared to the GG-DKF.

Chapter 4

Practical Implementation of Joint Probabilistic Data Association

4.1 Introduction

JPDA is one of the most widely-accepted MTT paradigms and is proved to generate MMSE estimates [52]. The advantage of utilising JPDA for MTT lies in that it considers all possible measurement-to-target associations in track update. This property makes JPDA strongly robust against the clutters and other undesired environmental disturbances. However, practical implementation of JPDA using low-cost sensors requires careful adjustment of the following two sensor-related issues:

- Computational complexity;
- Prior knowledge on detection probability and clutter rate.

Compared to the powerful MHT, JPDA can achieve reasonable results with lower computational burden. However, implementing JPDA still requires to enumerate all feasible joint association hypotheses to find the marginal association probability. Note that the marginal probability is then used to perform moment matching to merge the posterior Gaussian mixture into a single Gaussian form. Computing the marginal probabilities itself is known as a #P-complete problem [107, 108], thereby leading to the intractability for a large-scale tracking problem. Note that reducing the computational burden of JPDA is of paramount importance for low-cost sensors as they are generally battery-powered with limited computational capability.

Except for the computational issue, the JPDA filter, similar to most MTT algo-

rithms, requires the knowledge of detection probability and clutter rate in implementation. These two parameters are of great importance for the JPDA filter as they determine the final estimation performance. Significant mismatches of these two key parameters could result in erroneous tracking outputs. Although the detection probability, dependent on the sensors and the scenarios, can be tested via extensive offline experiments, it might not be cost-effective and it is typically unavailable for low-cost sensors. Furthermore, under dynamic environmental conditions or different target detection approaches, the clutter rate generated by the same sensor might change in a great deal: for example, in image-based target tracking, measurements are extracted from various detection algorithms, ranging from simple background subtraction [83] to complicated deep learning approach [84]. Obviously, the clutter rate and detection probability are different for different detection methods. Therefore, how to accommodate the unknown detection probability and clutter rate is of critical importance in practice.

This chapter investigates the problem of practical implementation of JPDA using low-cost sensors and suggests new variants of JPDA to address this problem. The main contributions of this chapter are twofold:

- For practical feasibility, this chapter suggests an efficient marginalisation algorithm using stochastic sampling techniques. The key idea is to consider each joint association hypothesis as a random variable, which is subject to a stationary distribution. Based on this concept, a Gibbs sampler is constructed to draw the joint association hypothesis samples and hence the marginal association probability can be easily approximated by the event occurrence. This provides polynomial-time approximation and makes it feasible to implement JPDA for large-scale tracking scenarios. Unlike previous *ad hoc* approximations [109, 110, 56, 2], the proposed Gibbs-JPDA is theoretically proved to guarantee global convergence and the upper bound on the convergence rate is also derived to support its application. This means that the Gibbs-JPDA ensures guaranteed performance, which is different from heuristic approximations.
- To relax the usual assumption on the full knowledge of detection probability and clutter rate, this chapter proposes a new variant of JPDA, termed as MB-JPDA, with capability of dynamic estimation of detection probability and clutter rate. The proposed MB-JPDA algorithm is derived by integrating a Gibbs-JPDA filter and a multi-Bernoulli filter. This new algorithm utilises a closed feedback loop structure based on the property of the posterior of the

JPDA estimations. More specifically, the multi-target JPDA filter leverages the information of estimated detection probability and clutter rate provided by the multi-Bernoulli filter. In the meantime, the multi-Bernoulli filter uses the estimation outputs of the main JPDA tracker since the posterior of JPDA estimations can be modelled by a multi-Bernoulli RFS.

4.2 System Models

This section provides necessary preliminaries of several important mathematical models to facilitate the analysis in the following sections.

4.2.1 Multi-Target State and Measurement Model

The sets of target states and measurements received at scan k are, respectively, defined as

$$\mathcal{X}_k = \{\mathbf{x}_k^1, \dots, \mathbf{x}_k^{N_k}\}, \quad \mathcal{Z}_k = \{\mathbf{z}_k^0, \mathbf{z}_k^1, \dots, \mathbf{z}_k^{M_k}\} \quad (4.1)$$

where N_k denotes the number of targets at scan k , $\mathbf{x}_k^i \in \mathbb{R}^n$ the i th target state at scan k , M_k the number of measurements received at scan k , $\mathbf{z}_k^j \in \mathbb{R}^m$ with $j \neq 0$ the j th measurement received at scan k , $\mathbf{z}_k^0 \in \mathbb{R}^m$ the dummy measurement for convenient representation of miss detection. For convenience, we make the following general assumptions, which are widely-accepted in MTT problems.

Assumption 4.1. *Each target can generate at most one measurement and each measurement can originate from at most one target. Each target-generated measurement is independent of each other and is detected with probability P_D with measurement likelihood $p(\mathbf{z}|\mathbf{x})$.*

Assumption 4.2. *The clutter distribution is assumed to be unknown a priori and is thus considered as Poisson distribution. Clutters or false alarms are modelled by a local Poisson point process (PPP) with intensity $\lambda_{FA}(\mathbf{z}) = N_{FA}/V$ with N_{FA} being the average number of clutters of one scan and V being the sensor volume.*

4.2.2 Target Existence Model

In MTT, there is no prior information on the source of received measurements. That is, each measurement may be spurious, i.e., false alarm, or from one existing/new

target. For this reason, both true tracks, representing real targets, and false tracks are generated and updated. The number of targets at time instant k is also a random variable N_k in a MTT problem: one target might suddenly disappear or appear in the sensor's field-of-view. Due to these facts, a track confirmation and deletion logic is essential in MTT to confirm the majority of true tracks and terminates most false tracks.

For simplicity, many MTT algorithms assume the number of targets is known *a priori* for data association and leverages a simple heuristic M/N logic for track management. This approach is straightforward, but requires extensive parameter tunings for different scenarios since it never considers the track quality. Hence, dynamical target management is more beneficial and easier to adapt to different tracking tasks. For this reason, we utilise a PPP model with intensity $\lambda_B(\mathbf{x})$, similar to [59], for target birth and perform track confirmation/deletion by the existence probability thresholding approach [54]. Within this framework, measurements that originate from extraneous source, either new targets or false alarms, can be modelled by a PPP with intensity $\lambda_E = \lambda_B(\mathbf{x}_b) p(\mathbf{z}|\mathbf{x}_b) P_D + \lambda_{FA}(\mathbf{z})$, where $\mathbf{x}_b$ denotes the possible state vector of birth targets. For target birth, the following general assumption is utilised in this chapter.

Assumption 4.3. *The number of new targets is assumed to be unknown a priori and is thus considered as Poisson distribution. New targets are modeled by a local Poisson point process with intensity $\lambda_B = N_B/V$ with N_B being the average number of new targets.*

Let χ_k^i represent the event of the i th target existence. If $p(\chi_k^i|\mathcal{Z}_k)$ is larger than a upper threshold α_1 , then the i th track is confirmed; once $p(\chi_k^i|\mathcal{Z}_k)$ is below a certain lower bound α_2 , the i th track is immediately deleted. For the case where $\alpha_2 \leq p(\chi_k^i|\mathcal{Z}_k) \leq \alpha_1$, the i th track is tentative and requires more information to confirm or delete. The prediction or time evolution of χ_k^i can be formulated by the Markov Chain One model [54] as

$$p(\chi_k^i|\mathcal{Z}_{k-1}) = P_S p(\chi_{k-1}^i|\mathcal{Z}_{k-1}) \quad (4.2)$$

where P_S denotes the target survival probability.

Remark 4.1. *Note that the utilisation of PPP birth model for target existence estimation is similar to the idea of JIPDA [54, 55]. However, the difference lies in that the JIPDA filter assumes a stationary or constant intensity of extraneous*

sources, which only include false alarms, i.e., $\lambda_E = \lambda_{FA}(\mathbf{z})$, while the proposed JPDA with PPP birth model dynamically estimate this intensity. Clearly, leveraging the PPP birth model is beneficial and more realistic in initialisation and calculating the marginal association probability.

4.3 Joint Probabilistic Data Association Filter

JPDA forms tracks out of the marginal measurement-to-track association distributions, and thus is a track-oriented approach. This section will first briefly introduce the basic concepts of JPDA and the issues associated with practical implementations will then be analysed.

4.3.1 Track Update of JPDA

JPDA aims to calculate the marginalised association probability for estimation update by enumerating all possible joint hypotheses. A feasible joint hypothesis in JPDA is defined as one possible measurement-to-target mapping such that:

- Each measurement (except for the dummy one) is assigned to at most one target;
- Each target is uniquely assigned to a measurement.

Let vector $\boldsymbol{\theta}_k = \{\theta_k^i\}^T$, $i \in \{1, 2, \dots, N_{k-1} + M_k\}$, denote the joint association hypothesis. For each pre-existed target $i \in \{1, 2, \dots, N_{k-1}\}$, define $\theta_k^i \in \{0, 1, \dots, M_k\}$ as the association hypothesis, and let $\theta_k^i = j$ denote the event that the j th measurement is originated from the i th target with $\theta_k^i = 0$ representing miss-detection event. We create a new track for each measurement $j \in \{1, 2, \dots, M_k\}$ at scan k , and the association events for these new targets are defined by $\theta_k^{N_{k-1}+j} \in \{N_{k-1} + 1, \dots, N_{k-1} + M_k\}$. That is, if target $N_{k-1} + j$ is associated with the j th measurement, then $\theta_k^{N_{k-1}+j} = N_{k-1} + j$. Under the assumption that each single association event is independent, the posterior of each target can be modelled by a mixture term as

$$p(\mathbf{x}_k^i | \chi_k^i, \mathcal{Z}_k) = \sum_{\theta_k^i} p(\mathbf{x}_k^i | \theta_k^i, \chi_k^i, \mathcal{Z}_k) p(\theta_k^i | \chi_k^i, \mathcal{Z}_k) \quad (4.3)$$

Generally, the number of mixture terms increases exponentially due to the data association uncertainty. Since every mixture term requires a Kalman filter for its

propagation, it is necessary to control the number of mixture terms from the computational burden standpoint. The standard JPDA simply approximates the posterior mixture PDF by a single distribution through moment-preserving method. More specifically, the state correction $\mathbf{x}_{k|k}^i$ of the i th target and its corresponding covariance $\mathbf{P}_{k|k}^i$ are obtained as

$$\begin{aligned}\mathbf{x}_{k|k}^i &= \sum_{j=0}^{M_k} \beta_j^i \mathbf{x}_{k|k}^{i,j} \\ \mathbf{P}_{k|k}^i &= \sum_{j=0}^{M_k} \beta_j^i \left\{ \mathbf{P}_{k|k}^{i,j} + \left(\mathbf{x}_{k|k}^{i,j} - \mathbf{x}_{k|k}^i \right) \left(\mathbf{x}_{k|k}^{i,j} - \mathbf{x}_{k|k}^i \right)^T \right\}\end{aligned}\quad (4.4)$$

where $\mathbf{x}_{k|k}^{i,j}$ denotes the target estimation by associating the j th measurement to the i th target, $\mathbf{P}_{k|k}^{i,j}$ the corresponding covariance, $\beta_j^i = p(\theta_k^i = j | \chi_k^i, \mathcal{Z}_k)$ the existence-conditioned marginal association probability that the j th measurement is associated with the i th target.

Note that $\{\mathbf{x}_{k|k}^{i,j}, \mathbf{P}_{k|k}^{i,j}\}$ can be extracted from $p(\mathbf{x}_k^i | \chi_k^i, \mathcal{Z}_k)$ by standard Kalman filter algorithm based on association event $\theta_k^i = j$ and the existence-conditioned marginal association probability $p(\theta_k^i | \chi_k^i, \mathcal{Z}_k)$ can be calculated using Bayesian rule as

$$\begin{aligned}p(\theta_k^i | \chi_k^i, \mathcal{Z}_k) &= \frac{p(\theta_k^i, \chi_k^i | \mathcal{Z}_k)}{p(\chi_k^i | \mathcal{Z}_k)} \\ &= \frac{p(\chi_k^i | \theta_k^i, \mathcal{Z}_k) p(\theta_k^i | \mathcal{Z}_k)}{p(\chi_k^i | \mathcal{Z}_k)}\end{aligned}\quad (4.5)$$

where the posterior probability of target existence $p(\chi_k^i | \mathcal{Z}_k)$ is determined by the law of total probability as

$$p(\chi_k^i | \mathcal{Z}_k) = \sum_{\theta_k^i} p(\theta_k^i, \chi_k^i | \mathcal{Z}_k) \quad (4.6)$$

The hypothesis-conditioned existence probability $p(\chi_k^i | \theta_k^i, \mathcal{Z}_k)$ is given by [54]

$$p(\chi_k^i | \theta_k^i, \mathcal{Z}_k) \propto \begin{cases} \frac{p(\chi_k^i | \mathcal{Z}_{k-1}) (1 - P_D)}{1 - p(\chi_k^i | \mathcal{Z}_{k-1}) + p(\chi_k^i | \mathcal{Z}_{k-1}) (1 - P_D)}, & \theta_k^i = 0 \\ 1, & \theta_k^i = j \\ \frac{P_D \lambda_{BP}(\mathbf{z}_k^j | \mathbf{x}_b)}{\lambda_{FA} + P_D \lambda_{BP}(\mathbf{z}_k^j | \mathbf{x}_b)}, & \theta_k^{N_{k-1}+j} = N_{k-1} + j \end{cases} \quad (4.7)$$

Now, the remaining part for track update is how to obtain the marginal association probability $p(\theta_k^i | \mathcal{Z}_k)$. Theoretically, the marginal association probability $p(\theta_k^i | \mathcal{Z}_k)$ can be calculated by finding all the feasible joint association hypotheses as

$$p(\theta_k^i = j | \mathcal{Z}_k) = \sum_{\theta_k^i \in \theta_k = j} p(\theta_k | \mathcal{Z}_k) \quad (4.8)$$

where the posterior distribution of the joint association hypothesis $p(\theta_k | \mathcal{Z}_k)$ is given by

$$p(\theta_k | \mathcal{Z}_k) \propto \left(\prod_{i=1}^{N_{k-1}} \varphi(\theta_k^i) \right) \left(\prod_{(i,i') \in E} \varphi_c(\theta_k^i, \theta_k^{i'}) \right) \quad (4.9)$$

where $\varphi(\theta_k^i)$ denotes the un-normalised probability distribution of single association hypothesis θ_k^i given by

$$\varphi(\theta_k^i) \propto \begin{cases} 1 - P_{DP}(\chi_k^i | \mathcal{Z}_{k-1}), & \theta_k^i = 0 \\ P_{DP}(\chi_k^i | \mathcal{Z}_{k-1}) p(\mathbf{z}_k^j | \mathbf{x}_{k|k-1}^i), & \theta_k^i = j \\ \lambda_{FA} + P_D \lambda_{BP}(\mathbf{z}_k^j | \mathbf{x}_b), & \theta_k^i = N_{k-1} + j \end{cases} \quad (4.10)$$

and

$$\varphi_c(\theta_k^i, \theta_k^{i'}) = \begin{cases} 0, & \theta_k^i = \theta_k^{i'} > 0 \\ 1, & \text{otherwise} \end{cases} \quad (4.11)$$

$$E = \{(i, i') | \exists i \in [N_{k-1}], i' \in [N_{k-1}], i \neq i'\} \quad (4.12)$$

which ensure that one measurement (except for the dummy measurement) can only be allocated to one target.

Algorithm 4.1 Joint Probabilistic Data Association Filter

- 1: Predict N_{k-1} existing targets using target dynamics
 - 2: Update each track with all possible measurements by Kalman filter
 - 3: Calculate the marginal probability $p(\theta_k^i | \mathcal{Z}_k)$
 - 4: Calculate the posterior existence probability $p(\chi_k^i | \mathcal{Z}_k)$
 - 5: Calculate the existence-conditioned marginal probability $p(\theta_k^i | \chi_k^i, \mathcal{Z}_k)$
 - 6: Use the moment matching (4.4) to update N_{k-1} existing targets
 - 7: Generate M_k new targets (one for each measurement)
 - 8: Perform track confirmation and deletion based using the existence probability $p(\chi_k^i | \mathcal{Z}_k)$
-

The implementation details of the JPDA filter is summarised in Algorithm 4.1. The outputs of JPDA at each time instant are the posterior existence probability $p(\chi_k^i | \mathcal{Z}_k)$ and posterior state estimate $(\mathbf{x}_{k|k}^i, \mathbf{P}_{k|k}^j)$ extracted from $p(\mathbf{x}_k^i | \chi_k^i, \mathcal{Z}_k)$. Note that the number of targets can be easily estimated by counting the confirmed

tracks. Therefore, JPDA provides a complete framework for MTT and track management.

4.3.2 Implementation Issues

Computational Complexity

Note from Eq. (4.8) that finding the exact marginal association probability $p(\theta_k^i | \mathcal{Z}_k)$ requires full enumeration of all possible joint associate hypotheses. This means that the complexity of JPDA is proportional to the number of joint association hypotheses, which is a function of the number of targets and measurements. In the worst case, i.e., without gating, the total number of joint association hypotheses given M_k measurements and N_{k-1} targets is

$$\begin{aligned}
 N_{joint} &= \underbrace{\sum_{i=0}^{\min\{M_k, N_{k-1}\}}}_{\text{Sum over the number of targets observed}} \underbrace{\binom{N_{k-1}}{i}}_{\text{Choose which targets are observed}} \underbrace{\binom{M_k}{i}}_{\text{Choose which measurements aren't clutter}} \underbrace{i!}_{\text{Assign the measurements to the targets}} \\
 &= \sum_{i=0}^{\min\{M_k, N_{k-1}\}} \underbrace{\frac{N_{k-1}! M_k!}{i! (N_{k-1} - i)! (M_k - i)!}}_{a_i}
 \end{aligned} \tag{4.13}$$

Notice that

$$\frac{a_{i+1}}{a_i} = \frac{(-i + N_{k-1})(-i + M_k)}{i + 1} \tag{4.14}$$

The total number of joint association hypotheses N_{joint} can be determined in terms of a hypergeometric function as

$$N_{joint} = {}_2F_0(-N_{k-1}, -M_k; 1) \tag{4.15}$$

Table 4.1 summarises the total number of joint association hypotheses given different number of measurements and targets for some small-scale problems. Even though the gating technique can be utilised to split large-scale association problems into several small-scale sub-problems, it can be noted from Table 4.1 that finding all possible joint association hypotheses becomes prohibitive even for some small-scale problems ($M_k = 5$ and $N_{k-1} = 5$ for example). Therefore, exact brute force solution through full enumeration (4.8) is computationally intractable except for a few simple

cases because of the combinatorial nature. Fig. 4.1 shows the mean average running time of standard JPDA for a typical scenario. From this figure, it can be noted that the time complexity of JPDA increases exponentially with the increasing of the number of targets. This prohibits the application of JPDA using low-cost sensors and motivates us to develop an alternative way to calculate the marginal association probability $p(\theta_k^i | \mathcal{Z}_k)$ in a tractable manner.

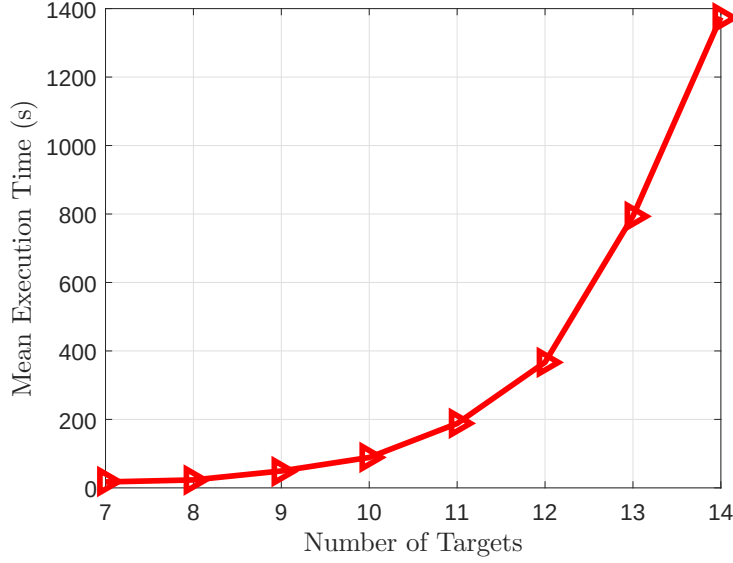


Figure 4.1: Mean average running time of standard JPDA for a typical scenario.

Table 4.1: Total number of joint association hypotheses.

$N_{k-1} \backslash M_k$	1	2	3	4	5	6
1	1	2	3	4	5	6
2	1	3	7	13	21	31
3	1	4	13	34	73	136
4	1	5	21	73	209	501
5	1	6	31	136	501	1546

Prior Knowledge on P_D and λ_{FA}

It follows from Eq. (4.10) that the implementation of JPDA requires the knowledge on detection probability P_D and clutter rate λ_{FA} . Significant mismatches of these two key parameters could result in erroneous tracking outputs. To see this, we reformulate Eq. (4.9) as

$$p(\boldsymbol{\theta}_k | \mathcal{Z}_k) \propto f_m \times f_t \times f_e \quad (4.16)$$

where f_m , f_t and f_e refer to the terms related to miss-detection, confirmed targets and external sources (false alarms and new targets), respectively, as

$$f_m \propto \prod_{i \in [N_{k-1}], \theta_k^i = 0} [1 - P_D p(\chi_k^i | \mathcal{Z}_{k-1})] \quad (4.17)$$

$$f_t \propto \prod_{i \in [N_{k-1}], \theta_k^i = j} [P_D p(\chi_k^i | \mathcal{Z}_{k-1}) p(\mathbf{z}_k^j | \mathbf{x}_{k|k-1}^i)] \quad (4.18)$$

$$f_e \propto \prod_{\theta_k^{N_{k-1}+j} = N_{k-1}+j} \left[\lambda_{FA}(\mathbf{z}_k^j) + \sum_{s=1}^{N_b} P_D \lambda_B(\mathbf{x}_b^s) p(\mathbf{z}_k^j | \mathbf{x}_b^s) \right] \quad (4.19)$$

where $\mathbf{x}_b^s$ denotes sth possible state vector of new birth targets and N_b represents the total number of possible initialisations for new targets. Since, under any association hypothesis, a measurement will either be associated to an existing target or an external source, Eq. (4.16) can be reformulated as

$$p(\boldsymbol{\theta}_k | \mathcal{Z}_k) \propto f_m \times f_d \quad (4.20)$$

where

$$f_d \propto \prod_{i \in [N_{k-1}], \theta_k^i = j} \left[\frac{P_D p(\chi_k^i | \mathcal{Z}_{k-1}) p(\mathbf{z}_k^j | \mathbf{x}_{k|k-1}^i)}{\lambda_{FA}(\mathbf{z}_k^j) + \sum_{s=1}^{N_b} P_D \lambda_B(\mathbf{x}_b^s) p(\mathbf{z}_k^j | \mathbf{x}_b^s)} \right] \quad (4.21)$$

The effect of the detection probability P_D on the estimation are two folds. On the one hand, it follows from Eqs. (4.17) and (4.21) that the detection probability P_D influences the relative ratio or weighting between the target miss detection f_m and target detection f_d terms. Therefore, artificially lower detection probability will give more penalty on the target-absence f_m , leading to the delayed track maintenance. However, lower detection probability might be useful in holding targets through temporary fades. On the other hand, the spatial density of the external source term, i.e., $\lambda_E = \sum_{s=1}^{N_b} \lambda_B(\mathbf{x}_b^s) p(\mathbf{z} | \mathbf{x}_b^s) P_D + \lambda_{FA}(\mathbf{z})$, also reveals that the detection probability also affects the track initialisation process. Lower detection probability will force longer tentative tracks before a confirmation or deletion decision is made based on the score logic. This, in turn, will increase the workload of the tracking filter as more tracks remain tentative, which is especially severe in a high-clutter scenario.

From Eq. (4.21), one can note that the clutter rate λ_{FA} also plays an important role in the JPDA algorithm. Higher clutter rate will give more penalty on the external

source term, i.e., false alarms or new targets, $\lambda_{FA}(\mathbf{z}_k^j) + \sum_{s=1}^{N_b} P_D \lambda_B(\mathbf{x}_b^s) p(\mathbf{z}_k^j | \mathbf{x}_b^s)$ than the target-detection term $p(\chi_k^i | \mathcal{Z}_k) p(\mathbf{z}_k^j | \mathbf{x}_k^i) P_D$. Therefore, artificially higher clutter rate provides an estimation result that more tracks will be considered as false alarms by the tracking system. Similarly, lower clutter rate usually generates overestimation of the real targets.

Consequently, significant mismatch of either detection probability P_D or clutter rate λ_{FA} will result in highly biased estimation. However, both P_D and λ_{FA} are difficult or not cost-effective to obtain in practice, especially for low-cost sensors. Therefore, it is imperative to design an improved JPDA algorithm that can accommodate this issue.

4.4 Approximated Marginalisation by Stochastic Gibbs Sampling

This section will propose a stochastic sampling approach to approximate the marginal association probability for efficient implementation of JPDA.

4.4.1 Sampling Joint Association Hypotheses

The key idea that this chapter proposes is to consider each joint association hypothesis $\boldsymbol{\theta}_k$ as a random variable that satisfies a distribution $\pi(\boldsymbol{\theta}_k)$. To this end, we construct a Markov chain whose state space is the set of all feasible joint association hypotheses with stationary distribution as the posterior distribution of joint association hypothesis, i.e., $p(\boldsymbol{\theta}_k | \mathcal{Z}_k)$. To ensure that the joint association hypothesis with higher probability are more easily to be sampled, it is natural to construct the sampling proposal $\pi(\boldsymbol{\theta}_k)$ proportional to its corresponding joint posterior probability, i.e.,

$$\pi(\boldsymbol{\theta}_k) \propto \left(\prod_{i=1}^{N_{k-1}} \varphi(\theta_k^i) \right) \left(\prod_{(i,i') \in E} \varphi_c(\theta_k^i, \theta_k^{i'}) \right) \quad (4.22)$$

Using the sampling proposal $\pi(\boldsymbol{\theta}_k)$, we could generate sufficiently enough samples of $\boldsymbol{\theta}_k$. From the samples, it is straightforward to approximate the marginal joint association probability by the sample occurrence. The issue is that direct sampling from proposal distribution (4.22) is very difficult as enumerating all possible joint association hypotheses is impossible for real-time applications. Therefore, we de-

velop a sampling-based marginalisation algorithm using Gibbs sampling, similar to [111]. Gibbs sampling is a stochastic method for Bayesian inference to approximate the posterior multivariate probability distribution in a polynomial time [112]. Gibbs sampler is a special case of the Metropolis-Hastings algorithm with an acceptance probability of one. The main advantage is that it is simpler to recursively sample from a conditional distribution than to sample directly from the joint distribution itself. More specifically, the transition kernel from one joint association hypothesis $\boldsymbol{\theta}_k = [\theta_k^1, \dots, \theta_k^{N_{k-1}}]^T$ to another one $\bar{\boldsymbol{\theta}}_k = [\bar{\theta}_k^1, \dots, \bar{\theta}_k^{N_{k-1}}]^T$ is given by

$$\pi(\bar{\boldsymbol{\theta}}_k | \boldsymbol{\theta}_k) = \prod_{m=1}^{N_{k-1}} \pi_m(\bar{\boldsymbol{\theta}}_k^m | \bar{\theta}_k^1, \dots, \bar{\theta}_k^{m-1}, \theta_k^{m+1}, \dots, \theta_k^{N_{k-1}}) \quad (4.23)$$

where π_m can be obtained from Eq. (4.22) using Bayesian rule as

$$\begin{aligned} & \pi_m(\bar{\boldsymbol{\theta}}_k^m | \bar{\theta}_k^1, \dots, \bar{\theta}_k^{m-1}, \theta_k^{m+1}, \dots, \theta_k^{N_{k-1}}) \\ &= \frac{\pi(\bar{\theta}_k^1, \dots, \bar{\theta}_k^m, \theta_k^{m+1}, \dots, \theta_k^{N_{k-1}})}{\pi(\bar{\theta}_k^1, \dots, \bar{\theta}_k^{m-1}, \theta_k^{m+1}, \dots, \theta_k^{N_{k-1}})} \\ &\propto \pi(\bar{\theta}_k^1, \dots, \bar{\theta}_k^m, \theta_k^{m+1}, \dots, \theta_k^{N_{k-1}}) \\ &= \left(\varphi(\vartheta_k^m) \prod_{(m,i') \in E} \varphi_c(\vartheta_k^m, \vartheta_k^{i'}) \right) \left(\prod_{i \neq m} \varphi(\vartheta_k^i) \prod_{(i,i') \in E, i \neq m} \varphi_c(\vartheta_k^i, \vartheta_k^{i'}) \right) \\ &\propto \varphi(\vartheta_k^m) \prod_{(m,i') \in E} \varphi_c(\vartheta_k^m, \vartheta_k^{i'}) \end{aligned} \quad (4.24)$$

where $\vartheta_k^i = \bar{\theta}_k^i, i \in \{1, \dots, m\}$, $\vartheta_k^i = \theta_k^i, i \in \{m+1, \dots, N_{k-1}\}$.

For a valid joint association hypothesis, we have $\varphi_c(\theta_k^m, \theta_k^{i'}) = 1$. Substituting this expression into Eq. (4.24) yields

$$\pi_m(\bar{\boldsymbol{\theta}}_k^m | \bar{\theta}_k^1, \dots, \bar{\theta}_k^{m-1}, \theta_k^{m+1}, \dots, \theta_k^{N_{k-1}}) \propto \varphi(\vartheta_k^m) \quad (4.25)$$

‘Proportion to’ in Eq. (4.25) highlights the dependence of individual conditional distribution on ϑ_k^m , while rest parts are formed as the normalisation constant. Given the joint association hypothesis $\boldsymbol{\theta}_k$, another hypothesis $\bar{\boldsymbol{\theta}}_k$ can then be obtained by

recursive sampling according to the following individual conditional distributions

$$\begin{aligned}
\bar{\theta}_k^1 &\sim \pi_1 \left(\bar{\theta}_k^1 \mid \theta_k^2, \dots, \theta_k^{N_{k-1}} \right) \\
&\vdots \\
\bar{\theta}_k^m &\sim \pi_m \left(\bar{\theta}_k^m \mid \bar{\theta}_k^1, \dots, \bar{\theta}_k^{m-1}, \theta_k^{m+1}, \dots, \theta_k^{N_{k-1}} \right) \\
&\vdots \\
\bar{\theta}_k^{N_{k-1}} &\sim \pi_{N_{k-1}} \left(\bar{\theta}_k^{N_{k-1}} \mid \bar{\theta}_k^1, \dots, \bar{\theta}_k^{N_{k-2}} \right)
\end{aligned} \tag{4.26}$$

Once an enough number of samples generated, the marginal association probability is approximated by occurrence. The detailed pseudo code of the proposed Gibbs sampling-based marginalisation algorithm is summarised in Algorithm 4.2. Notice that samples generated at early iterations, known as the burn-in phase, usually cannot represent the target distribution and need to be discarded.

Algorithm 4.2 Marginalisation by Gibbs sampling

Input: Previous target estimation, received measurements, allowable samples N_{gibbs} , burn-in samples $N_{burn-in}$

Output: Marginal association probability $p(\theta_k^i \mid \chi_k^i, \mathcal{Z}_k)$

```

1:  $iter \leftarrow 1$  ▷ set the initial iteration counter as one
2:  $N_{\theta_j^i} \leftarrow 0, j \in \{1, 2, \dots, N_{k-1} + M_k\}$  ▷ set the event counter as zero
3:  $\theta_k^{(1)} \leftarrow \mathbf{0}_{N_{k-1}}$  ▷ set the initial state for the Gibbs sampler
4: while  $iter < N_{gibbs}$  do
5:    $iter \leftarrow iter + 1$ 
6:   for  $m = 1 : N_{k-1}$  do
7:     Recursive sampling according to Eq. (4.26)
8:   end for
9:    $\theta_k^{(iter)} = [\theta_k^1, \dots, \theta_k^{N_{k-1}}]^T$  ▷ one Gibbs sample
10:  if  $iter > N_{burn-in}$  then
11:    if  $\theta_k^i = j$  then
12:       $N_{\theta_j^i} \leftarrow N_{\theta_j^i} + 1$ 
13:    end if
14:  end if
15: end while
16:  $p(\theta_k^i = j \mid \chi_k^i, \mathcal{Z}_k) = N_{\theta_j^i} / (N_{gibbs} - N_{burn-in})$ 
17: return  $p(\theta_k^i = j \mid \chi_k^i, \mathcal{Z}_k)$ 

```

Remark 4.2. *Similar to Metropolis-Hastings sampling, the Gibbs sampling might become inefficient in exploring the space with high dimensionality, i.e., extremely large number of targets, because of the random-walk behaviour [113]. Under this condition, the Hamiltonian Monte Carlo sampling could be utilised as an alterna-*

tive way to obtain the samples in a more efficient way. Compared to Gibbs sampling, the Hamiltonian Monte Carlo sampling has faster convergence speed for high-dimensional target distribution, although the price of single iteration is higher [113].

Remark 4.3. From Algorithm 4.2, it can be noted that the time complexity of the Gibbs sampler is proportional to the number of samples N_{gibbs} . Given fixed number of Gibbs samples, the complexity of the proposed Gibbs sampling-based marginalisation is proportional to the number of targets, since the size of one joint association hypothesis θ_k is N_{k-1} . This means that the computational complexity of the proposed approach is linear with respect to the problem size and therefore it is scalable.

4.4.2 Convergence of Gibbs Sampler

After constructing the Markov chain, it is necessary to prove that the generated Markov chain asymptotically converges to its invariant distribution and this property is established in the following Theorem.

Theorem 4.1. *Given any initial feasible joint event, the distribution of Gibbs samples (4.23) asymptotically converges to the target distribution (4.22) with an exponential rate as*

$$|\pi^n(\bar{\theta}_k | \theta_k) - \pi(\bar{\theta}_k)| \leq (1 - 2\beta)^{\lfloor n/2 \rfloor} \quad (4.27)$$

where $\pi^n(\bar{\theta}_k | \theta_k)$ denotes the n th power of transition kernel $\pi(\bar{\theta}_k | \theta_k)$, i.e., the transition probability from θ_k to $\bar{\theta}_k$ via n steps, $\beta = \min_{\theta_k} \pi^2(\bar{\theta}_k | \theta_k) \in (0, 0.5]$ the least likely two-step transition probability.

Proof. In general, the convergence of finite-state Markov chain is guaranteed by its irreducibility and regularity. The irreducibility of a Markov chain is quantified in terms of the possibility that one state has capability to transfer to another state within finite step. And the regularity of a Markov chain can be checked by the positivity of the entries of some finite power of its transition matrix.

Since every target can share the dummy measurement, i.e., $\varphi_c(0, \theta_k^i) = 1$, it follows from Eq. (4.24) that

$$\begin{aligned} \pi(\mathbf{0}_{N_k} | \theta_k) &\propto \prod_{m=1}^{N_{k-1}} \varphi(0) > 0 \\ \pi(\theta_k | \mathbf{0}_{N_k}) &\propto \prod_{m=1}^{N_{k-1}} \varphi(\theta_k^m) > 0 \end{aligned} \quad (4.28)$$

Then, the two-step transition kernel from any $\boldsymbol{\theta}_k$ to any $\bar{\boldsymbol{\theta}}_k$ satisfies

$$\begin{aligned}\pi^2(\bar{\boldsymbol{\theta}}_k | \boldsymbol{\theta}_k) &= \sum_{\zeta} \pi(\bar{\boldsymbol{\theta}}_k | \zeta) \pi(\zeta | \boldsymbol{\theta}_k) \\ &> \pi(\bar{\boldsymbol{\theta}}_k | \mathbf{0}_{N_k}) \pi(\mathbf{0}_{N_k} | \boldsymbol{\theta}_k) > 0\end{aligned}\tag{4.29}$$

This implies that the Markov chain $\{\boldsymbol{\theta}_k^{(t)}\}_{t=1}^{\infty}$ generated by the Gibbs sampler is irreducible and recurrent, and therefore the Markov chain will asymptotically converge to its invariant distribution, e.g., the posterior of the joint association hypothesis, by the ergodic theorem [114]. Next, applying Lemma B.2, presented in Appendix B, to $\pi^2(\bar{\boldsymbol{\theta}}_k | \boldsymbol{\theta}_k)$ gives

$$\begin{aligned}\max_{\boldsymbol{\theta}_k} \pi^{2n}(\bar{\boldsymbol{\theta}}_k | \boldsymbol{\theta}_k) - \min_{\boldsymbol{\theta}_k} \pi^{2n}(\bar{\boldsymbol{\theta}}_k | \boldsymbol{\theta}_k) &\leq (1 - 2\beta)^n \\ \lim_{n \rightarrow \infty} \max_{\boldsymbol{\theta}_k} \pi^{2n}(\bar{\boldsymbol{\theta}}_k | \boldsymbol{\theta}_k) &= \lim_{n \rightarrow \infty} \min_{\boldsymbol{\theta}_k} \pi^{2n}(\bar{\boldsymbol{\theta}}_k | \boldsymbol{\theta}_k) \geq \beta > 0\end{aligned}\tag{4.30}$$

Since Lemma B.1, presented in Appendix B, states that $\max_{\boldsymbol{\theta}_k} \pi^n(\bar{\boldsymbol{\theta}}_k | \boldsymbol{\theta}_k)$ is non-increasing and $\min_{\boldsymbol{\theta}_k} \pi^n(\bar{\boldsymbol{\theta}}_k | \boldsymbol{\theta}_k)$ is non-decreasing in n , Eq. (4.30) can be reformulated as

$$\begin{aligned}\max_{\boldsymbol{\theta}_k} \pi^n(\bar{\boldsymbol{\theta}}_k | \boldsymbol{\theta}_k) - \min_{\boldsymbol{\theta}_k} \pi^n(\bar{\boldsymbol{\theta}}_k | \boldsymbol{\theta}_k) &\leq (1 - 2\beta)^{\lfloor n/2 \rfloor} \\ \lim_{n \rightarrow \infty} \max_{\boldsymbol{\theta}_k} \pi^n(\bar{\boldsymbol{\theta}}_k | \boldsymbol{\theta}_k) &= \lim_{n \rightarrow \infty} \min_{\boldsymbol{\theta}_k} \pi^n(\bar{\boldsymbol{\theta}}_k | \boldsymbol{\theta}_k) > 0\end{aligned}\tag{4.31}$$

Due to the asymptotic convergence property of the proposed Markov chain, we have

$$\pi(\bar{\boldsymbol{\theta}}_k) = \lim_{n \rightarrow \infty} \max_{\boldsymbol{\theta}_k} \pi^n(\bar{\boldsymbol{\theta}}_k | \boldsymbol{\theta}_k) = \lim_{n \rightarrow \infty} \min_{\boldsymbol{\theta}_k} \pi^n(\bar{\boldsymbol{\theta}}_k | \boldsymbol{\theta}_k)\tag{4.32}$$

Since $\pi(\bar{\boldsymbol{\theta}}_k)$ lies between the minimum and maximum $\pi^n(\bar{\boldsymbol{\theta}}_k | \boldsymbol{\theta}_k)$ for any given state $\boldsymbol{\theta}_k$, Eq. (4.27) can be directly ensured. $\square$

Remark 4.4. *Theorem 4.1 reveals that the convergence error of the proposed Gibbs sampler is upper bounded by a known constant. This means that the approximation by Gibbs sampling is performance-guaranteed regardless of the setting of the number of the burn-in samples. This is a main difference between Gibbs-JPDA and other heuristic approximations [109, 110, 56, 2].*

Remark 4.5. *Typically, there is no rule-of-thumb or analytically way to set the number of burn-in phase samples. However, due to the exponential convergence rate, the burn-in phase of the proposed Gibbs sampler is short. Even though the number of burn-in samples is empirically set, its influence on the marginalisation is*

ignorable since the generated Gibbs samples are not used for inference the stationary distribution. This will be empirically analysed in the simulation part.

Remark 4.6. Theorem 4.1 shows that, given any feasible joint association hypothesis as the initial state $\boldsymbol{\theta}_k^{(1)}$, the generated Markov chain $\{\boldsymbol{\theta}_k^{(t)}\}_{t=1}^{\infty}$ exponentially converges to the posterior of the joint association hypothesis. Due to the convergence property, one can easily select a feasible joint association hypothesis as the initial state for Gibbs sampler. For example, one can choose the trivial joint association hypothesis that all targets are assumed to be miss-detected as the initial state of the Gibbs sampler, e.g., $\boldsymbol{\theta}_k^{(1)} = \mathbf{0}_{N_{k-1}}$.

4.5 JPDA with Unknown Detection Probability and Clutter Rate

This section proposes a new variant of JPDA filter to adjust the unknown detection probability and clutter rate by leveraging the multi-Bernoulli filter, thus termed as MB-JPDA.

4.5.1 Detection Probability and Clutter Rate Estimation

The recently proposed multi-Bernoulli filter [115, 79, 80] utilises the RFS theory to recursively propagate the posterior of multi-target state. This filter is known as a parametrised approximation of the Bayesian multi-target recursion. At each time instant k , the posterior PDF of the multi-target state is characterised as a multi-Bernoulli RFS. More specifically, each target is modelled as a single Bernoulli RFS, which is fully described by the existence probability $r_k^i = p(\chi_k^i | \mathcal{Z}_k)$ and the probability density $p_k(\mathbf{x}_k^i)$. A Bernoulli RFS is an empty set with probability $1 - r_k^i$ and has only one element, whose distribution is described by $p(\mathbf{x}_k^i)$, with probability r_k^i . The multi-Bernoulli RFS is a simple union of N_k single independent Bernoulli RFSs and thus can be completely characterised by the parameter set $\{r_k^i, p(\mathbf{x}_k^i)\}_{i=1}^{N_k}$. The recursive multi-Bernoulli filter is given by two steps as [115, 79]:

- **Prediction.** If the posterior multi-target density at time $k-1$ is characterised as a multi-Bernoulli RFS as $\pi_{k-1|k-1} = \left\{ r_{k-1|k-1}^i, p_{k-1|k-1}(\mathbf{x}_{k-1}^i) \right\}_{i=1}^{N_{k-1}}$, then,

the prediction of the multi-target density is also a multi-Bernoulli as

$$\pi_{k|k-1} = \{r_{B,k}^i, p_{B,k}(\mathbf{x}_k^i)\}_{i=1}^{N_{B,k}} \cup \{r_{k|k-1,i}, p_{k|k-1}(\mathbf{x}_{k-1}^i)\}_{i=1}^{N_{k-1}} \quad (4.33)$$

where the first part denotes the multi-Bernoulli RFS of the birth model and the second part represents the prediction of existing targets with

$$r_{k|k-1}^i = r_{k-1|k-1}^i \langle p_{k-1|k-1}(\mathbf{x}_{k-1}^i), P_S \rangle \quad (4.34)$$

$$p_{k|k-1}(\mathbf{x}_{k-1}^i) = \frac{\langle f(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i), p_{k-1|k-1}(\mathbf{x}_{k-1}^i) P_S \rangle}{\langle p_{k-1|k-1}(\mathbf{x}_{k-1}^i), P_S \rangle} \quad (4.35)$$

where $\langle \alpha, \beta \rangle \triangleq \int \alpha(\mathbf{x}) \beta(\mathbf{x}) d\mathbf{x}$.

- **Update.** If the predicted multi-target density at time k is characterised as a multi-Bernoulli RFS as $\pi_{k|k-1} = \{r_{k|k-1}^i, p_{k|k-1}(\mathbf{x}_k^i)\}_{i=1}^{N_{k-1}}$, then, the update of the multi-target density is also a multi-Bernoulli as

$$\pi_k = \{r_{L,k}^i, p_{L,k}(\mathbf{x}_k^i)\}_{i=1}^{N_{k-1}} \cup \{r_{U,k}(\mathbf{z}), p_{U,k}(\cdot, \mathbf{z})\}_{\mathbf{z} \in \mathcal{Z}_k} \quad (4.36)$$

where the legacy (miss detection) component $\{r_{L,k}^i, p_{L,k}(\mathbf{x}_k^i)\}_{i=1}^{N_{k-1}}$ is given by

$$r_{L,k}^i = r_{k|k-1}^i \frac{1 - \langle p_{k|k-1}(\mathbf{x}_{k-1}^i), P_D \rangle}{1 - r_{k|k-1}^i \langle p_{k|k-1}(\mathbf{x}_{k-1}^i), P_D \rangle} \quad (4.37)$$

$$p_{L,k}(\mathbf{x}_k^i) = p_{k|k-1}(\mathbf{x}_k^i) \frac{1 - P_D}{1 - \langle p_{k|k-1}(\mathbf{x}_k^i), P_D \rangle} \quad (4.38)$$

and the measurement updated component is given by

$$r_{U,k}(\mathbf{z}) = \frac{\sum_{i=1}^{N_{k-1}} \frac{r_{k|k-1}^i (1 - r_{k|k-1}^i) \langle p_{k|k-1}(\mathbf{x}_k^i), p(\mathbf{z} | \mathbf{x}_k^i) P_D \rangle}{(1 - r_{k|k-1}^i \langle p_{k|k-1}(\mathbf{x}_k^i), P_D \rangle)^2}}{\lambda_{FA}(\mathbf{z}) + \sum_{i=1}^{N_{k-1}} \frac{r_{k|k-1}^i \langle p_{k|k-1}(\mathbf{x}_k^i), p(\mathbf{z} | \mathbf{x}_k^i) P_D \rangle}{1 - r_{k|k-1}^i \langle p_{k|k-1}(\mathbf{x}_k^i), P_D \rangle}} \quad (4.39)$$

$$p_{U,k}(\mathbf{x}_k^i, \mathbf{z}) = \frac{\sum_{i=1}^{N_{k-1}} \frac{r_{k|k-1}^i P_D}{1 - r_{k|k-1}^i} p_{k|k-1}(\mathbf{x}_k^i) p(\mathbf{z} | \mathbf{x}_k^i)}{\sum_{i=1}^{N_{k-1}} \frac{r_{k|k-1}^i}{1 - r_{k|k-1}^i} \langle p_{k|k-1}(\mathbf{x}_k^i), P_D \rangle} \quad (4.40)$$

The advantage of the multi-Bernoulli filter is that it utilises the mathematically sound RFS theory and thus directly avoids the data association process in multi-

target tracking, leading to computational efficiency. However, the target identity or the target label cannot be maintained during the filtering recursion. This means that the multi-Bernoulli filter cannot be used to support the target label management. Therefore, if the label information is required for practice, the application of multi-Bernoulli filter is limited. Fortunately, as our aim is to estimate the clutter rate as well as the detection probability using the multi-Bernoulli filter, there is no need to manage the clutter identity.

To apply the multi-Bernoulli filter to estimate the clutter rate, we consider each false alarm as a target with its own transition model, birth model and death model [80]. This consideration is motivated by the fact that the target and the clutter have totally different dynamics and thus can be treated separately in estimation. The true targets and clutters in recursive filtering are therefore labelled by a discrete space $\mathcal{I} = \{0, 1\}$, where 1 is the label of real targets and 0 is for clutters. To further accommodate the detection probability estimation, we augment the state space as $\mathcal{X}_a = \mathcal{P} \times \mathcal{X} \times \mathcal{I}$, where $\mathcal{P}$ denotes the space of detection probability and is defined on the continuous domain $[0, 1]$. Here, the notation $\times$ represents the Cartesian product. In Bayesian filtering, both prediction and update steps require the integration over the state space. Note that the new state space is discrete due to the introduced labelling space $\mathcal{I}$, the integration on this space can then be split into two different parts. Define $\mathcal{X}_p = \mathcal{P} \times \mathcal{X}$, then the integration over space $\mathcal{X}_a$ can be obtained as

$$\int_{\mathcal{X}_a} f(\mathbf{x}_a) d\mathbf{x}_a = \int_{\mathcal{X}_p} f_0(\mathbf{x}_p) d\mathbf{x}_p + \int_{\mathcal{X}_p} f_1(\mathbf{x}_p) d\mathbf{x}_p \quad (4.41)$$

where f_1 and f_0 , respectively, refer to the models of targets and clutters.

In reality, the detection probability is related to the target state. This fact makes the system state propagation/prediction of the augmented system intractable. To address this problem, we further assume that the detection probability is independent of the target state, i.e.,

$$\begin{aligned} f_0(P_D, \mathbf{x}) &= f_0(P_D) f_0(\mathbf{x}) \\ f_1(P_D, \mathbf{x}) &= f_1(P_D) f_1(\mathbf{x}) \end{aligned} \quad (4.42)$$

Substituting this expression into Eq. (4.41) gives

$$\int_{\mathcal{X}_a} f(\mathbf{x}_a) d\mathbf{x}_a = \int_{\mathcal{X}} \int_{\mathcal{P}} f_0(P_D) f_0(\mathbf{x}) dP_D d\mathbf{x} + \int_{\mathcal{X}} \int_{\mathcal{P}} f_1(P_D) f_1(\mathbf{x}) dP_D d\mathbf{x} \quad (4.43)$$

In general, the clutter dynamics and detection probability are highly dependent on the types of sensors. If we have some prior information regarding the sensors, we can utilise such information to build the clutter dynamics and detection probability model. Otherwise, general models can be leveraged, e.g., Gaussian distribution and random walk. By substituting the augmented state space into the original multi-Bernoulli filter, we can readily estimate the number of clutters at one scan $\hat{N}_{FA}$ by the summation of all confirmed targets with label 0 and the clutter rate estimation $\hat{\lambda}_{FA}$ is then given by $\hat{\lambda}_{FA} = \hat{N}_{FA}/V$. As P_D is part of the system states, it can be directly obtained by the state estimation of multi-Bernoulli filter.

Remark 4.7. *Note that the target detection probability used in JPDA is typically piecewise constant and thus the independence assumption is reasonable in filter design. Furthermore, the validation presented in the following section reveals that the proposed algorithm works well in the presence of time-varying detection probability.*

Remark 4.8. *Note that one can also apply the augmented system into JPDA to adapt to unknown detection probability and clutter rate. However, the inherent data association process of JPDA will make practical implementation intractable.*

4.5.2 MB-JPDA Algorithm

The general information flow of the proposed MB-JPDA algorithm is shown in Fig. 4.2. At every time instant, we utilise the proposed Gibbs-JPDA as the main tracker to provide the multi-target state and identity estimation. The required detection probability and clutter rate are obtained from an one-step multi-Bernoulli filter. By incorporating the PPP model into JPDA, the posterior of JPDA estimation can be modelled as a multi-Bernoulli RFS and thus the estimated output of the JPDA provides feedback to the multi-Bernoulli filter for its initialisation at every time instant. The advantage of the proposed parallel filtering scheme is that one can fully exploit the strengths of both approaches: using the JPDA for track management and obtaining fast estimation of detection probability as well as clutter rate by the multi-Bernoulli filter. This property is different from some previous works, where the clutter rate estimation is decoupled from the multi-object estimation [85, 86, 87, 88].

Remark 4.9. *Since additional multi-Bernoulli filter is utilised for detection probability and clutter rate estimation, the computational complexity of MB-JPDA is higher than that of the Gibbs-JPDA. However, the RFS-based multi-Bernoulli filter directly avoids the typical data association procedure and therefore MB-JPDA also runs in a polynomial time by using Gibbs sampling for marginalisation.*

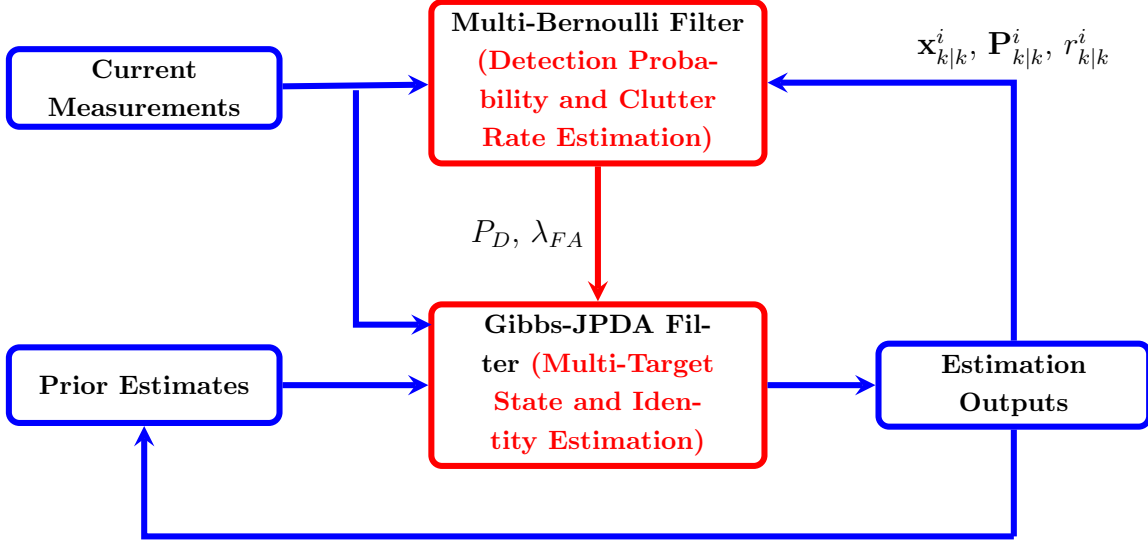


Figure 4.2: General information flow of MB-JPDA.

4.6 Numerical Simulations

In this section, the effectiveness of the proposed JPDA filtering algorithms is investigated through numerical simulations in a cluttered environment.

4.6.1 Simulation Setup

Our experiments explore a scenario, involving ten manoeuvring targets with different birth time. The state vector contains planar position and velocity. We use the well-known coordinated-turning (CT) model for target prediction. The CT model is defined as

$$\mathbf{x}_k^i = \mathbf{F}_{CT} \mathbf{x}_{k-1}^i + \mathbf{G} w_{k-1}^i \quad (4.44)$$

with

$$F_{CT} \triangleq \begin{bmatrix} 1 & \frac{\sin(\omega_k^i T_s)}{\omega_k^i} & 0 & -\frac{1-\cos(\omega_k^i T_s)}{\omega_k^i} \\ 0 & \cos(\omega_k^i T_s) & 0 & -\sin(\omega_k^i T_s) \\ 0 & \frac{1-\cos(\omega_k^i T_s)}{\omega_k^i} & 1 & \frac{\sin(\omega_k^i T_s)}{\omega_k^i} \\ 0 & \sin(\omega_k^i T_s) & 0 & \cos(\omega_k^i T_s) \end{bmatrix}, \quad \mathbf{G} \triangleq \begin{bmatrix} T_s^2/2 \\ T_s \\ T_s^2/2 \\ T_s \end{bmatrix} \quad (4.45)$$

where $T_s = 1s$ denotes the sampling period, and $w_k^i \sim \mathcal{N}(\cdot; 0, \sigma_a^2)$ the Gaussian process noise with $\sigma_a = 15m/s^2$. The turning rate ω_k^i is modelled by a random walk

as

$$\omega_k^i = \omega_{k-1}^i + \eta_{k-1}^i \quad (4.46)$$

where $\eta_k^i \sim \mathcal{N}(\cdot; 0, \sigma_\omega^2)$ is the Gaussian process noise with $\sigma_\omega = 3^\circ/s$.

It is assumed that we use a radar for multiple targets tracking. The nonlinear range and bearing measurements are generated with detection probability $P_D = 0.9$ as

$$\mathbf{z}_k^i = \begin{bmatrix} \sqrt{(p_{x,k}^i)^2 + (p_{y,k}^i)^2} \\ \arctan\left(\frac{p_{y,k}^i}{p_{x,k}^i}\right) \end{bmatrix} + \mathbf{v}_k^i \quad (4.47)$$

where $(p_{x,k}^i, p_{y,k}^i)$ denotes the 2D inertial position of the i th target, $\mathbf{v}_k \sim \mathcal{N}(\cdot; 0, \mathbf{R}_k^i)$ the Gaussian measurement noise with $\mathbf{R}_k^i = \text{diag}(\sigma_r^2, \sigma_\theta^2)$ with $\sigma_r = 5m$ and $\sigma_\theta = 1^\circ$. The clutter is assumed to be uniformly distributed in the surveillance region with its number being Poisson with $N_{FA} = 10$ average returns at each scan. Snapshots of one sample trajectory for each target and 'transformed' measurements are depicted in Fig. 4.3. Note that the term 'transformed' measurements here means the nonlinear range and angle measurements are transformed into Cartesian coordinate for depiction only.

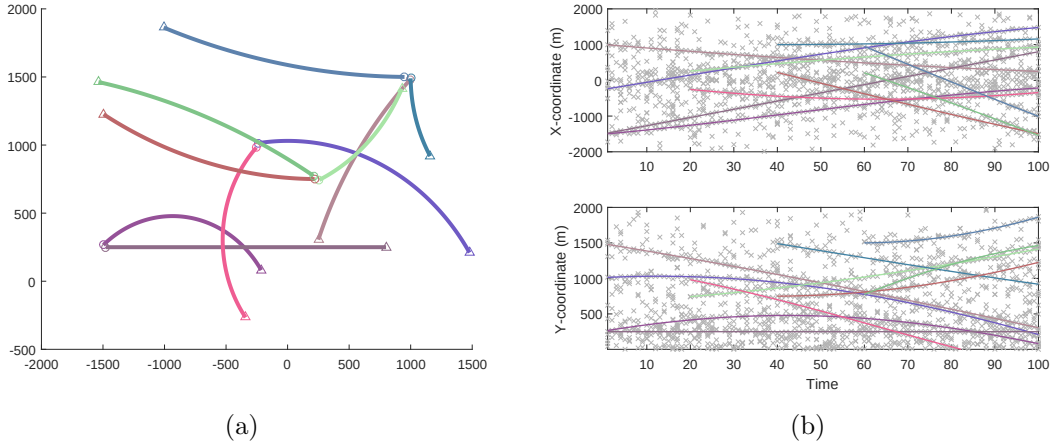


Figure 4.3: Snapshots of the considered scenario. (a) Ground truth with 'circle' being birth point and 'triangle' being death point. (b) One-dimensional target position and measurements with grey 'x' being 'transformed' measurements and colour lines being targets.

Tentative track is confirmed if the existence probability satisfies $p(\chi_k^i | \mathcal{Z}_k) \geq 0.8$ and a confirmed track is deleted immediately once $p(\chi_k^i | \mathcal{Z}_k) \leq 0.1$. For implementing Gibbs sampling, the number of burn-in samples is set as $N_{burn-in} = 5$ and the number of Gibbs samples is chosen as $N_{gibbs} = 30$.

In order to evaluate the performance of the proposed algorithms with different conditions, one parameter is varied while others are set as their aforementioned default values. The simulations are obtained over 500 Monte Carlo runs for each parameter setting.

4.6.2 Performance Metric

The optimal sub-pattern assignment (OSPA) distance metric [116] is considered here for overall performance evaluation, namely, cardinality and position estimation errors. Let $\mathcal{X} \triangleq \{\mathbf{x}_1, \dots, \mathbf{x}_m\}$ and $\mathcal{Y} \triangleq \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$ be the position estimation set and true target position set, respectively. The cardinality of these two sets are m and n , respectively. Then, for $c > 0$ and $1 \leq p < \infty$, the OSPA distance $d_p^c(\mathcal{X}, \mathcal{Y})$ is defined as [116]

$$d_p^c(\mathcal{X}, \mathcal{Y}) \triangleq \begin{cases} \left[\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^m d^c(\mathbf{x}_i, \mathbf{y}_{\pi(i)})^p + c^p (n - m) \right) \right]^{1/p}, & m \leq n \\ d_p^c(\mathcal{Y}, \mathcal{X}), & m > n \end{cases} \quad (4.48)$$

where Π_n denotes the set of all permutations on $\{1, 2, \dots, n\}$ for any positive integer n . $d^c(\mathbf{x}_i, \mathbf{y}_{\pi(i)})$ is the cut-off Euclidean distance between two vectors as $d^c(\mathbf{x}_i, \mathbf{y}_{\pi(i)}) = \min(d(\mathbf{x}_i, \mathbf{y}_{\pi(i)}), c)$ with $d(\mathbf{x}_i, \mathbf{y}_{\pi(i)})$ being the Euclidean distance. The order parameter p determines the sensitivity of $d_p^c(\mathcal{X}, \mathcal{Y})$ in penalizing estimation outliers, while the cut-off parameter c determines the relative weighting of the penalties allocated to cardinality and localisation errors. In all simulations, these two parameters are set as $p = 1$, $c = 100$.

4.6.3 Performance of Gibbs-JPDA

Scalability and Robustness of Gibbs-JPDA

As can be noted from Eq. (4.13), the time complexity of standard JPDA is not scalable with respect to the problem size, i.e., number of targets and number of measurements. To show the scalability of using Gibbs sampling in marginalisation, Monte-Carlo simulations are performed with respect different number of targets $N_{target} = 5, 6, 7, 8, 9, 10$ and clutters $N_{FA} = 10, 20, 30, 40$ and the results are pro-

vided in Figs. 4.4 and 4.5¹. Tables 4.2 and 4.3 summarise the corresponding time averaged mean OSPA distance and running time. From Figs. 4.4 (b) and 4.5 (b), it can be noted that the time complexity of the proposed Gibbs-JPDA is linear with respect to the number of targets and the number of measurements. This confirms that the proposed algorithm runs in polynomial time and therefore is scalable. Unsurprisingly, the tracking performance of JPDA degrades with more targets. The reason is that the chance of data association failure rises with the increase in the number of targets, which has an adverse effect on the tracking accuracy. However, the discrepancy of mean OSPA distance does not differ much for different number of targets. This means that the proposed algorithm is capable of handling large tracking scenarios. High clutter rate is a typical characteristic of low-cost sensors, which would have adverse effect on the tracking performance. Intuitively, the increase of clutter rate will give rise to high possibility of data association failure, thus the degradation in tracking performance. This can be clearly observed in Fig. 4.5 (a), but the Gibbs-JPDA shows strongly robustness even against a high clutter rate, e.g., $N_{FA} = 40$. With the increasing of the average number of clutters per scan, the variation in the number of measurements also increases. This, in turn, results in the increase of the variation of processing time, as can be confirmed by Fig. 4.5 (b).

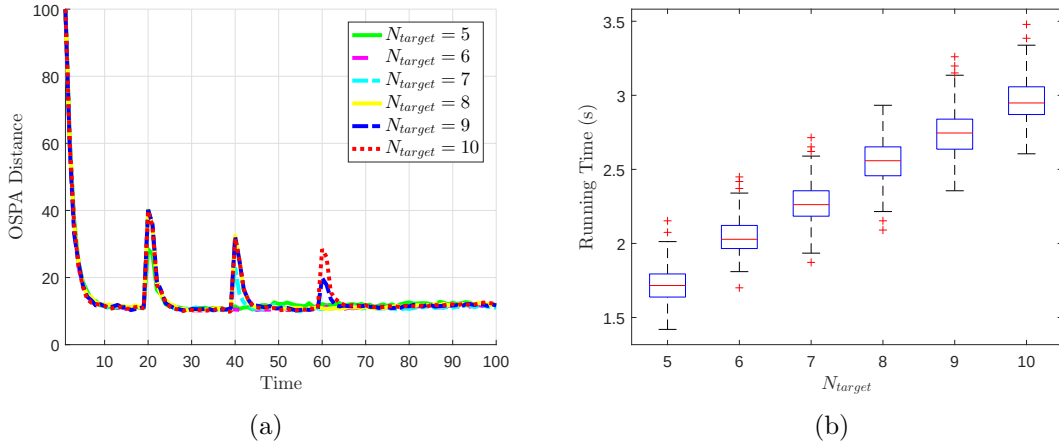


Figure 4.4: Monte-Carlo comparison results with respect to different number of targets. (a) Mean OSPA distance. (b) Running time.

It follows from Algorithm 4.2 that the time complexity of Gibbs sampler is also proportional to the number of samples and Theorem 4.1 reveals that the distribution

¹In plotting the running time, we leverage the boxplot to display the distribution of recorded data. The red line inside the blue box refers to the median value and the blue box stands for the 25%-75% region of record data. The upper and low black lines stand for the most extreme data points not considered as outliers, and the outliers are plotted individually using the red '+' symbol.

Table 4.2: Mean OSPA distance and running time w.r.t. number of targets.

N_{target}	5	6	7	8	9	10
OSPA distance	13.9488	13.5404	13.7047	14.2330	14.2979	14.5011
Running time	1.7180s	2.0437s	2.2723s	2.5540s	2.7468s	2.9646s

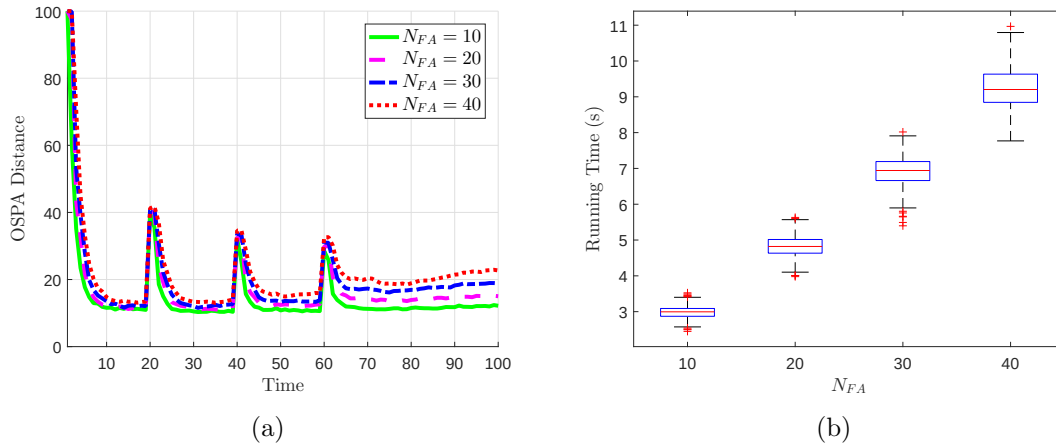


Figure 4.5: Monte-Carlo comparison results with respect to different number of clutters. (a) Mean OSPA distance. (b) Running time.

of Gibbs samples will asymptotically converge to the target distribution of joint association hypothesis. In order to investigate the effect of the parameter N_{gibbs} on tracking performance, Monte Carlo simulations are performed with respect to different numbers of Gibbs samples $N_{gibbs} = 10, 20, 30, 40$. The simulation results of OSPA distance and running time are depicted in Fig. 4.6 and the corresponding time-averaged performance is summarised in Table 4.4. From Fig. 4.6 and Table 4.4, it can be observed that increasing the number of samples generate better tracking performance in terms of OSPA distance at the sacrifice of computational time. This conforms with statement made in Theorem 4.1. However, when the number of samples is larger than a certain threshold, e.g., $N_{gibbs} \geq 20$ in the considered scenario, the performance improvement induced by the increasing of N_{gibbs} is not very obvious since enough samples are generated to represent the association hypotheses. Furthermore, Fig. 4.6 (b) also reveals that running time of Gibbs-JPDA increases linearly with respect to the number of samples.

As mentioned before, there is no rule-of-thumb to select the number of the burn-in samples is empirically set in the simulations. Fig. 4.7 studies the impact of number of burn-in samples on the performance of the proposed tracking algorithm and Table 4.5 summarises the time averaged mean OSPA distance and running time. From

Table 4.3: Mean OSPA distance and running time w.r.t. number of clutters.

N_{FA}	10	20	30	40
OSPA distance	14.6057	17.1644	19.4571	21.9547
Running time	2.9835s	4.8216s	6.9206s	9.2290s

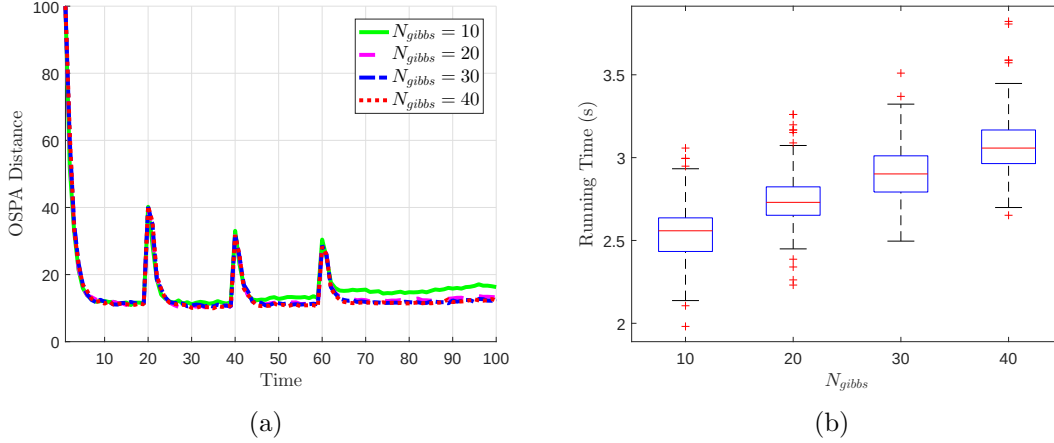


Figure 4.6: Monte-Carlo comparison results with respect to different number of Gibbs samples. (a) Mean OSPA distance. (b) Running time.

this table, it is clear that $N_{burn-in} = 5$ achieves the best performance in terms of estimation accuracy, but the tracking performance in terms of the OSPA distance under all considered conditions is comparable, as can be confirmed by Fig. 4.7 (a). This means that the proposed algorithm is robust against the variation of number of burn-in samples. Also, Fig. 4.7 (b) reveals that increasing the number of burn-in samples can slightly reduce the computational time. This can be attributed to the fact that less samples have been processed during the marginalisation. However, the discrepancy of running time among all tested cases is very small and can be ignorable.

Comparison with Other Approximated Marginalisation Algorithms

To further validate the proposed Gibbs-JPDA, simulations are performed with comparison to other approximated marginalisation algorithms. More specifically, the following four different algorithms are considered in the comparison:

- Algorithm 1: ENNJPDA [1];
- Algorithm 2: Modified cheap JPDA [2];
- Algorithm 3: Suboptimal JPDA [3];

Table 4.4: Mean OSPA distance and running time w.r.t. number of Gibbs samples.

N_{gibbs}	10	20	30	40
OSPA distance	16.3298	15.0163	14.8542	14.6621
Running time	2.5389s	2.7412s	2.9085s	3.0745s

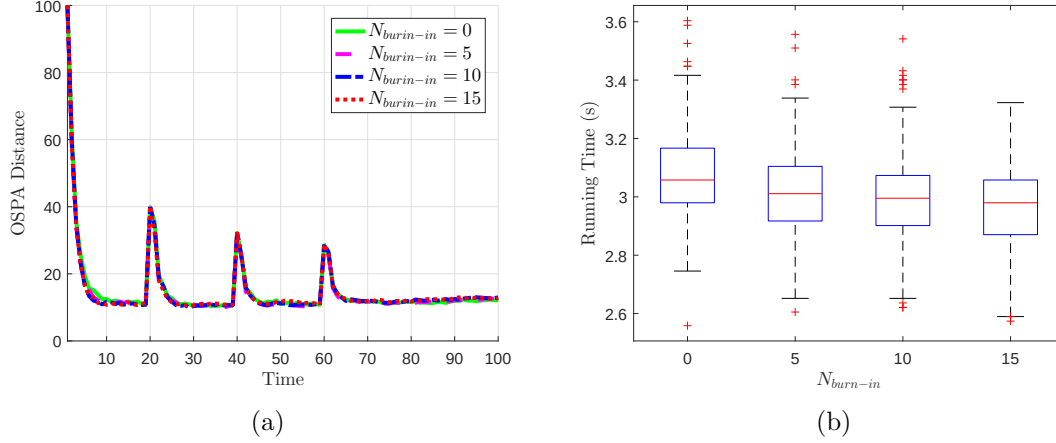


Figure 4.7: Monte-Carlo comparison results with respect to different number of burn-in samples. (a) Mean OSPA distance. (b) Running time.

- Algorithm 4: Gibbs JPDA.

The details of Algorithms 1-3 are summarised in Appendix D. The comparisons are carried out for three different cases $N_{target} = 10, 40, 80$. The Monte-Carlo simulation results for these three different cases are presented in Figs. 4.8-4.10. The corresponding time-averaged mean OSPA distance and execution time are summarised in Table 4.6. From Fig. 4.8 (a), it can be observed that Algorithms 3 and 4 have comparable performance and outperform the other two algorithms in terms of tracking accuracy for the scenario with 10 targets. However, Algorithm 4 has slightly higher time complexity than that of Algorithm 3, as demonstrated by Fig. 4.8 (b). With the increase the number of targets, the tracking accuracy of the proposed Gibbs-JPDA, i.e., Algorithm 4, significantly outperforms other algorithms, as confirmed by Figs. 4.9 (a) and 4.10 (a). This reveals that the proposed Gibbs-JPDA has strong robustness against the variation of the problem size. Although Algorithms 2-4 have similar level of time complexity for scenarios $N_{target} = 40$ and $N_{target} = 80$, the proposed Gibbs-JPDA has the smallest variation in execution time the lowest time complexity among all the tested algorithms, as shown in Figs. 4.9 (b) and 4.10 (b). These results clearly demonstrate that the proposed Gibbs-JPDA provides an efficient approximation of marginalisation with guaranteed performance for implementing JPDA.

Table 4.5: Mean OSPA distance and running time w.r.t. number of burn-in samples.

$N_{burn-in}$	0	5	10	15
OSPA distance	14.9362	14.5761	14.7741	14.8525
Running time	3.0756s	3.0159s	2.9876s	2.9602s

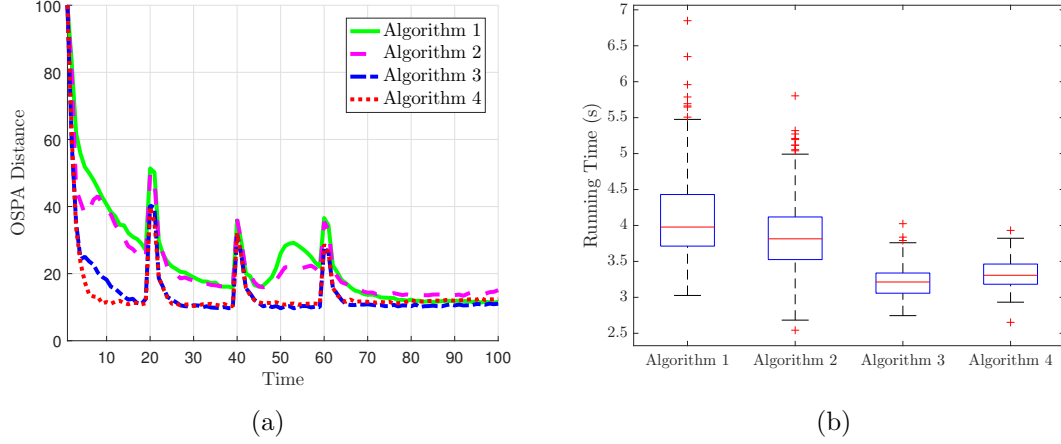


Figure 4.8: Monte-Carlo comparison results for a scenario with 10 targets. (a) Mean OSPA distance. (b) Running time.

4.6.4 Performance of MB-JPDA

In order to verify the effectiveness of the proposed MB-JPDA algorithm, the following two cases are considered in the simulations:

- Case 1 (stationary scenario): $P_D = 0.9$, $N_{FA} = 10$;
- Case 2 (dynamic scenario): P_D peaks at the sensor origin with $P_D = 0.95$ and exponentially decreases to 0.8 at the boundary of the sensing field-of-view. The average number of clutters is given by a time-varying profile as $N_{FA} = 10 + 5 \sin(0.1t)$.

For the purpose of comparison, we also perform the ideal JPDA filter with perfect knowledge of detection probability as well as clutter rate and the following four non-ideal JPDA filters:

- Non-JPDA 1: ideal P_D , $N_{FA} = 1/10N_{FA}^{real}$;
- Non-JPDA 2: ideal $P_D = 0.9$, $N_{FA} = 10N_{FA}^{real}$;
- Non-JPDA 3: $P_D = 0.6$, ideal N_{FA} ;
- Non-JPDA 4: $P_D = 0.3$, ideal N_{FA} .

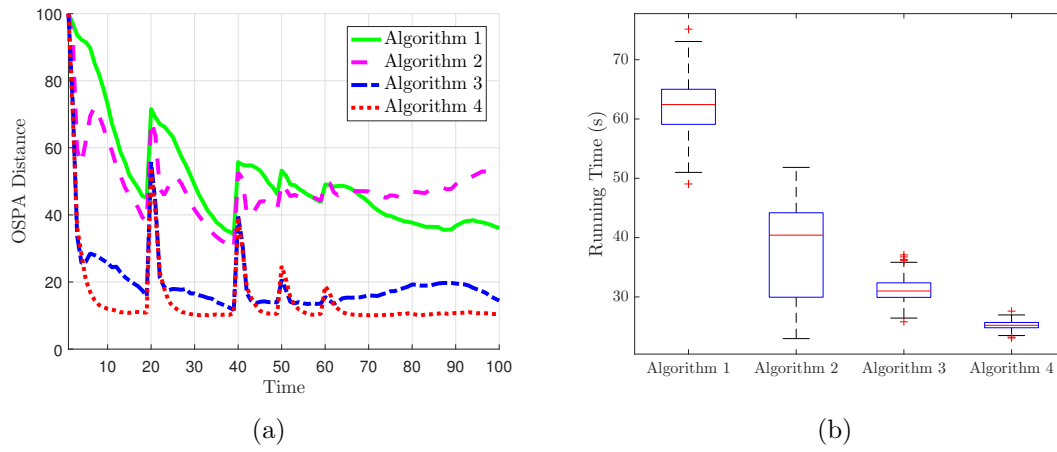


Figure 4.9: Monte-Carlo comparison results for a scenario with 40 targets. (a) Mean OSPA distance. (b) Running time.

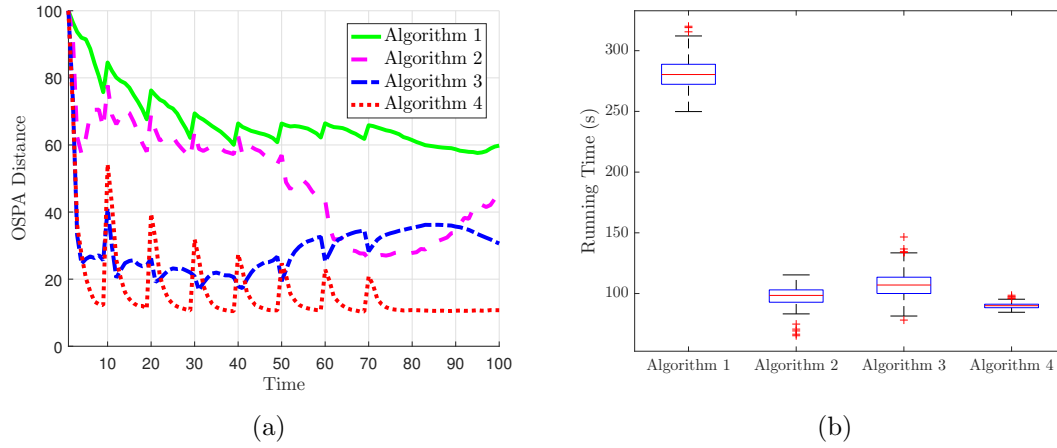


Figure 4.10: Monte-Carlo comparison results for a scenario with 80 targets. (a) Mean OSPA distance. (b) Running time.

where N_{FA}^{real} denotes the real average number of clutters per scan that is utilised in generate the scenario and N_{FA} refers to the average number of clutters per scan utilised in the JPDA.

The results of OSPA distance and cardinality estimation obtained by Monte-Carlo runs are shown in Figs. 4.11 and 4.12. The time-averaged mean OSPA distance obtained from different JPDAs in the simulations are summarised in Table 4.7. The peaks of mean OSPA distance in Fig. 4.11 are resulted from track confirmation for target birth. If the clutter rate is set lower than the real value, the JPDA filter overestimate the multi-target state and thus generating more 'confirmed' targets, leading to larger OSPA distance (e.g., non-ideal JPDA filter 1). If the clutter rate is set higher than the real value, the JPDA filter gives an underestimation the

Table 4.6: Comparison results of different approximated marginalisation algorithms.

N_{target}	Metric	Algorithm 1	Algorithm 2	Algorithm 3	Algorithm 4
10	OPSA distance	23.8797	22.6854	14.7758	14.6997
	Running time	4.1091s	3.8458s	3.2116s	3.3219s
40	OPSA distance	50.4952	48.7055	20.0683	14.9097
	Running time	62.3473s	37.9775s	31.1418s	25.2442s
80	OPSA distance	67.3621	49.2299	29.3754	16.5069
	Running time	280.9620s	97.8728s	107.5799s	89.9069s

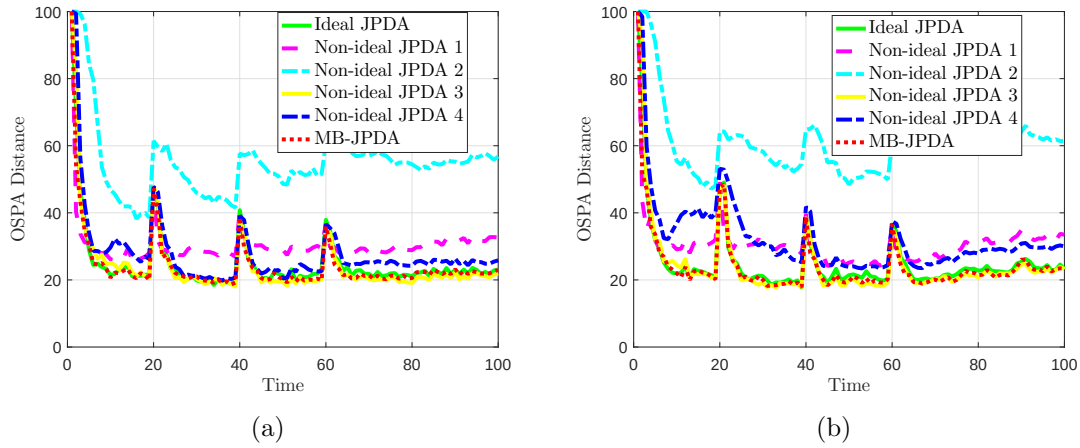


Figure 4.11: Monte-Carlo comparison results of mean OSPA distance for different cases. (a) Case 1. (b) Case 2.

multi-target state (e.g., non-ideal JPDA filter 2). Although the results, shown in Fig. 4.11, reveal that the JPDA filter is somewhat robust against the detection probability (e.g., non-ideal JPDA filter 3), low detection probability gives a delayed confirmation, as we stated earlier. Moreover, when the detection probability is highly mismatched with its real value (e.g., non-ideal JPDA filter 4), the estimation is unreliable. As a comparison, the results clearly verify that the proposed MB-JPDA filter has comparable performance as the ideal JPDA filter by utilising the multi-Bernoulli estimator in a feedback loop.

4.7 Conclusions

This chapter addresses the sensor-related implementation issues for practical application of the JPDA filter using low-cost sensors. Two main problems are considered in this chapter: (1) reducing the time complexity of finding the marginal association probability; and (2) alleviating the assumption on the full knowledge of detection

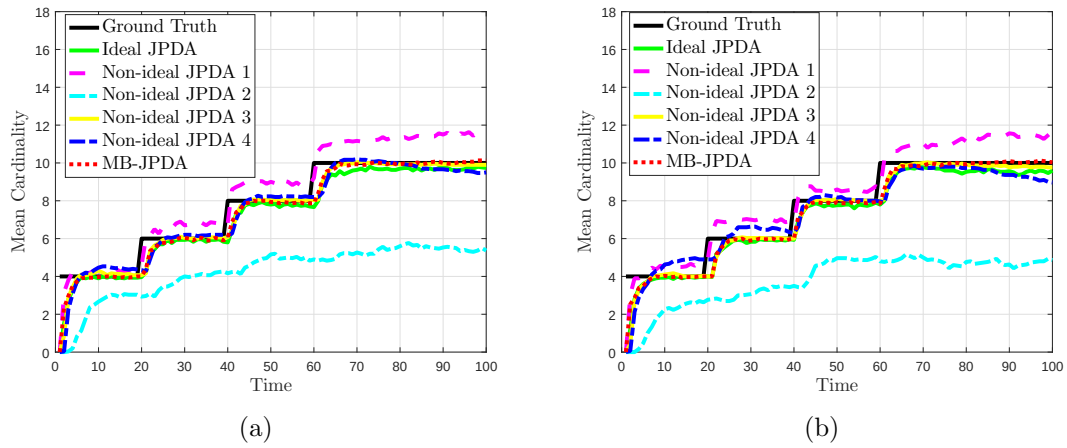


Figure 4.12: Monte-Carlo comparison results of mean cardinality estimation for different cases. (a) Case 1. (b) Case 2.

Table 4.7: Mean OSPA distance for different cases.

	Ideal JPDA	Non-ideal JPDA 1	Non-ideal JPDA 2	Non-ideal JPDA 3	Non-ideal JPDA 4	MB-JPDA
Case 1	24.0646	30.4442	55.1231	24.2637	27.9943	24.8453
Case 2	24.0454	30.7092	60.5923	24.4553	32.5549	25.0797

probability and clutter rate.

For the first issue, an efficient marginalisation approach is proposed by utilising stochastic Gibbs sampling. Each possible joint association hypothesis is considered as a random variable that can be drawn by stochastic Gibbs sampling and hence the marginal association probability can be easily approximated by the sample occurrence. This polynomial-time approximation makes it feasible to apply JPDA for multi-target tracking using low-cost sensors. Performance evaluation with extensive empirical simulations show that the proposed Gibbs-JPDA provides scalable and performance-guaranteed approximation, i.e., insensitive to the clutter rate and the variation of problem size, in marginalisation.

For the second issue, a new variant of JPDA, called MB-JPDA, is developed by incorporating with the multi-Bernoulli filter to accommodate the unknown detection probability and clutter rate. Based on the PPP birth model, the multi-target estimation output of Gibbs-JPDA can be characterised as a multi-Bernoulli RFS. This enables the application of the multi-Bernoulli filter in a feedback loop to estimate the unknown detection probability and clutter rate. Simulation results confirm that

the proposed algorithm exhibits the advantages of the Gibbs-JPDA filter and can correct the biased estimations induced by the unknown detection probability and clutter rate. Under the condition where detection probability and clutter rate are difficult to be obtained, the MB-JPDA filter can be a practical solution, providing performance comparable to the ideal JPDA filter.

Chapter 5

Information-Theoretic Joint Probabilistic Data Association

5.1 Introduction

It is known that the JPDA filter makes a soft decision in data association, i.e., finding the marginal association probability for track update. Due to the nature of the soft decision, the posterior of the i th target provided by JPDA can be modelled by a mixture term as

$$p(\mathbf{x}_k^i | \chi_k^i, \mathcal{Z}_k) = \sum_{\theta_k^i} p(\mathbf{x}_k^i | \theta_k^i, \chi_k^i, \mathcal{Z}_k) p(\theta_k^i | \chi_k^i, \mathcal{Z}_k) \quad (5.1)$$

Generally, the number of mixture terms increases exponentially due to the data association uncertainty. Since every mixture term requires a Kalman filter for its propagation, it is necessary to control the number of mixture terms from the computational burden standpoint. The standard JPDA simply approximates the posterior mixture PDF by a single distribution through moment-preserving method. More specifically, the state correction $\mathbf{x}_{k|k}^i$ of the i th target and its corresponding covariance $\mathbf{P}_{k|k}^i$ are obtained as

$$\begin{aligned} \mathbf{x}_{k|k}^i &= \sum_{j=0}^{M_k} \beta_j^i \mathbf{x}_{k|k}^{i,j} \\ \mathbf{P}_{k|k}^i &= \sum_{j=0}^{M_k} \beta_j^i \left\{ \mathbf{P}_{k|k}^{i,j} + \left(\mathbf{x}_{k|k}^{i,j} - \mathbf{x}_{k|k}^i \right) \left(\mathbf{x}_{k|k}^{i,j} - \mathbf{x}_{k|k}^i \right)^T \right\} \end{aligned} \quad (5.2)$$

As stated in [57], the simple moment-preserving approximation is accurate enough provided that the distance between Gaussian terms is far enough. If two targets are not well-spaced or in a highly cluttered environment (see Fig. 5.1 for example), the resulting Gaussian mixture exhibits multi-modality and therefore this approximation may destroy valuable information.

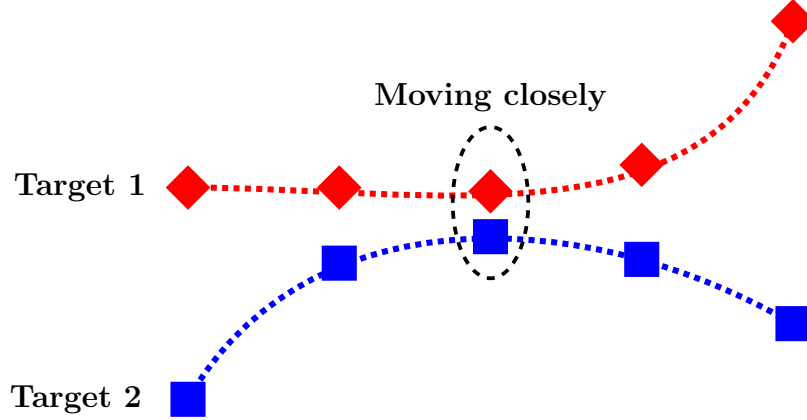


Figure 5.1: An example of closely-moving targets. The coloured diamonds and rectangles refer to the positions of two different targets at regular time instants.

To improve the approximation performance of Gaussian mixture approximation in JPDA, this chapter suggests a new information-theoretic JPDA algorithm that approximates the Gaussian mixture PDF in a more accurate way to improve the overall tracking performance of the JPDA filter through an information-theoretic approach. The main contribution of this chapter is:

- An information-theoretic approach is proposed to approximate the posterior Gaussian mixture PDF. This approximation is obtained by the minimisation of an information-theoretic metric, i.e., a weighted KLD. The uniqueness of the proposed algorithm is in choosing the weights for the mixture as the marginal association probabilities over the basic reverse KLD mixture model. Leveraging the weighted KLD enables the resultant PDF to approach the mode with the most probable association pair without completely pruning other information. Another benefit of using a weighted KLD as the cost function is that it enables a simple closed-loop solution, which largely resembles the concept of generalised CI and therefore provides robustness against cross-correlations.
- Under the condition of unity probability of target detection, it is proved that the proposed information-theoretic JPDA provides unbiased estimation. The lower and upper bounds on the estimation covariance are also analytically derived to support the validity of the proposed approach.

5.2 Information-Theoretic JPDA

This section will propose a novel information-theoretic JPDA by replacing the original moment-preserving mixture reduction method with an information-theoretic approach.

5.2.1 Information-Theoretic Mixture Approximation

Instead of simple moment-matching, we will propose an information-theoretic approach for Gaussian mixture reduction in JPDA. In general, there are many ways to quantify the proximity of two PDFs. Among them, the most widely-accepted metric, from the information-theoretic point of view, is the KLD or cross entropy [117]. Given two PDFs $p(\mathbf{x})$, $q(\mathbf{x})$, the KLD is defined as

$$\mathcal{D}_{KL}(p \| q) = \int p(\mathbf{x}) \ln \frac{p(\mathbf{x})}{q(\mathbf{x})} d\mathbf{x} \quad (5.3)$$

which reveals that $\mathcal{D}_{KL}(p \| q) \geq 0$ with equality if and only if $p(\mathbf{x}) = q(\mathbf{x})$. Therefore, the KLD can be viewed as a pseudo distance between two PDFs in a similar way as Euclidean distance between two points in an inertial frame. The reason why we call it a pseudo distance lies in its unsymmetrical property, that is, $\mathcal{D}_{KL}(p \| q) \neq \mathcal{D}_{KL}(q \| p)$.

As the KLD quantifies how close a PDF is to a candidate PDF. One can find a best approximation of a specific PDF in the sense of minimisation of KLD. Let $p(\mathbf{x})$ be the original PDF and $q(\mathbf{x})$ the candidate PDF that approximates the original one. In information theory, there are two different kinds of KLDs, forward KLD and reverse KLD, that we can seek to minimise them to obtain our solution $q(x)$. These two different KLDs are defined as [117]

$$\begin{aligned} \text{Forward KLD: } \mathcal{D}_{KL}(p \| q) &= \int p(\mathbf{x}) \ln \frac{p(\mathbf{x})}{q(\mathbf{x})} d\mathbf{x} \\ \text{Reverse KLD: } \mathcal{D}_{KL}(q \| p) &= \int q(\mathbf{x}) \ln \frac{q(\mathbf{x})}{p(\mathbf{x})} d\mathbf{x} \end{aligned} \quad (5.4)$$

The solution of $\min_q \mathcal{D}(p \| q)$ is known as moment projection, which finds the mean of $p(\mathbf{x})$ and overestimates the support of $p(\mathbf{x})$. In [118], the authors proposed a Gaussian mixture reduction method on the basis of the forward KLD, which was proved to only consider the merging operation for mixture reduction [119]. As a comparison,

minimisation of the reverse KLD results in a mode selection approximation [119]. In other words, this solution only keeps the strongest mode and prunes the mixture terms with low weightings. In JPDA filter, if several targets are moving closely, the Gaussian mixture terms tend to exhibit much higher multi-modality and thus the mode seeking property of the reverse KLD to prune the unreliable terms will generate improved estimation performance. This is also the reason why the drastic pruning method, ENNJPD, shows remarkable ability in closely-spaced scenarios. However, this drastic pruning also prunes some useful information and therefore is sensitive to clutter rate. In order to obtain improved estimation performance, we will seek to minimise the KLD to derive an overall optimal solution. Unfortunately, as stated in [119], the analytical solution of the problem of the minimisation of the reverse KLD for mixture distributions is intractable and one needs to resort to time-consuming numerical optimisation methods. Hence, numerical methods might not be suitable for online implementation of target tracking algorithm, especially for low-cost sensors. To maintain the feasibility, we will propose a weighted KLD that can be used to derive analytical solutions.

Similar to standard JPDA, our aim is to find an approximation of the Gaussian mixture by a single Gaussian PDF. To realise this, define the weighted KLD for the i th target as

$$\sum_{j=0}^{M_k} \beta_j^i \mathcal{D}_{KL} (p^i \| p_j^i) \quad (5.5)$$

where $p_j^i(\mathbf{x}^i) = \mathcal{N}(\mathbf{x}^i; \mathbf{x}_{k|k}^{i,j}, \mathbf{P}_{k|k}^{i,j})$ denotes the PDF representing the relationship between the i th target and the j th measurement, p^i the candidate approximated posterior PDF of the i th target. Then, our solution of Gaussian mixture approximation is given by the optimisation problem as

$$p_*^i = \arg \min_{p^i} \sum_{j=0}^{M_k} \beta_j^i \mathcal{D}_{KL} (p^i \| p_j^i) \quad (5.6)$$

Since the marginal association probability β_j^i quantifies the confidence level that the j th measurement comes from the i th target, β_j^i is leveraged as the weight/penalty for the difference between the candidate PDF p_i and the single association Gaussian term $p_j^i(x^i)$. This implies that the proposed cost function also considers matching with the strongest Gaussian mode, i.e., with the highest marginal association probability, to resolve the multi-modality issue when targets are moving very closely. The mathematical relationship between the original reverse KLD and the proposed cost

function is established in the following proposition.

Proposition 5.1. *The reverse KLD provides the lower bound of the proposed cost function, i.e.,*

$$\mathcal{D}_{KL} \left(p^i \left\| \sum_{j=0}^{M_k} \beta_j^i p_j^i \right\| \right) \leq \sum_{j=0}^{M_k} \beta_j^i \mathcal{D}_{KL} (p^i \| p_j^i) \quad (5.7)$$

where the equality holds if and only if $p_{j'}^i = p_j^i, \forall j' \neq j$.

Proof. According to the log-sum inequality, we have

$$\begin{aligned} \mathcal{D}_{KL} \left(p^i \left\| \sum_{j=0}^{M_k} \beta_j^i p_j^i \right\| \right) &= \int p^i(x^i) \ln \frac{p^i(x^i)}{\sum_{j=0}^{M_k} \beta_j^i p_j^i(x^i)} dx^i \\ &\leq \int \sum_{j=0}^{M_k} a_j p^i(x^i) \ln \frac{a_j p^i(x^i)}{\beta_j^i p_j^i(x^i)} dx^i \\ &= \sum_{j=0}^{M_k} a_j \int p^i(x^i) \ln \frac{p^i(x^i)}{p_j^i(x^i)} dx^i + \sum_{j=0}^{M_k} a_j \ln \frac{a_j}{\beta_j^i} \\ &= \sum_{j=0}^{M_k} a_j \mathcal{D}_{KL} (p^i \| p_j^i) + \sum_{j=0}^{M_k} a_j \ln \frac{a_j}{\beta_j^i} \end{aligned} \quad (5.8)$$

where $0 \leq a_j \leq 1$ and $\sum_{j=0}^{M_k} a_j = 1$, and the equality holds if and only if

$$\frac{a_{j'} p^i(x^i)}{\beta_{j'}^i p_{j'}^i(x^i)} = \frac{a_j p^i(x^i)}{\beta_j^i p_j^i(x^i)}, \quad \forall j' \neq j \quad (5.9)$$

Then, Proposition 5.1 can be proved by choosing $a_j = \beta_j^i$. □

Remark 5.1. *Proposition 5.1 indicates that the proposed cost function reduces to the reverse KLD once all Gaussian terms have the same mean and covariance. Fig. 5.2 provides a comparison example of different Gaussian mixture approximation approaches, where the solution of minimising the reverse KLD is obtained via numerical algorithms. From Fig. 5.2 (a), it can be clearly noted that minimising the reverse KLD and the proposed weighted KLD generate the same solution if all Gaussian terms have the same distribution. Comparing to the moment matching, i.e., minimising the forward KLD, optimising the reverse KLD provides the opportunity of mode selection and therefore can directly resolve the issue of multi-modality in JPDA. As an approximation of the reverse KLD, the proposed cost function provides the solution with approximate mode selection, as confirmed by Fig. 5.2.*

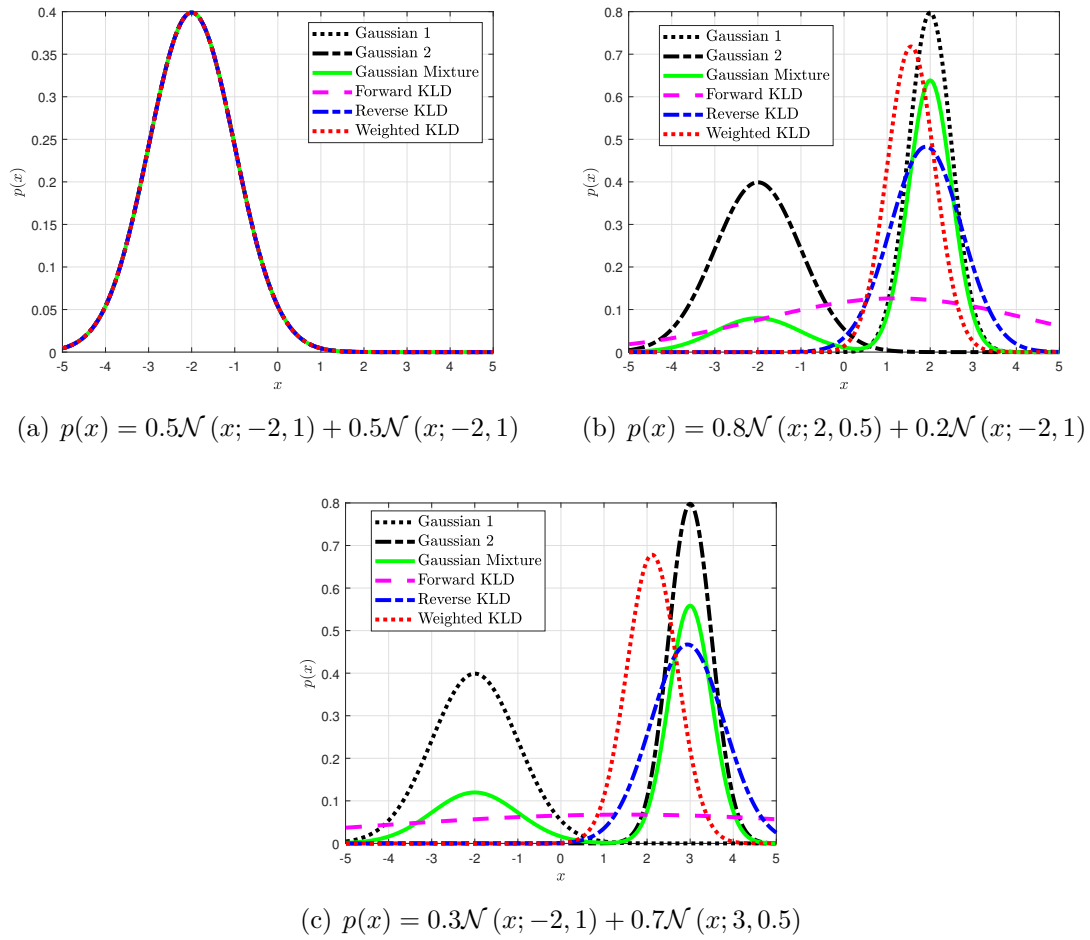


Figure 5.2: Comparison of different Gaussian mixture approximations.

Proposition 5.2. *The closed-form solution of optimisation problem (5.6) is given by*

$$p_*^i = \mathcal{N}(\mathbf{x}^i; \mathbf{x}_{k|k}^i, \mathbf{P}_{k|k}^i) \quad (5.10)$$

where

$$\begin{aligned} \mathbf{x}_{k|k}^i &= \mathbf{P}_{k|k}^i \sum_{j=0}^{M_k} \beta_j^i \left(\mathbf{P}_{k|k}^{i,j} \right)^{-1} \mathbf{x}_{k|k}^{i,j} \\ \mathbf{P}_{k|k}^i &= \left[\sum_{j=0}^{M_k} \beta_j^i \left(\mathbf{P}_{k|k}^{i,j} \right)^{-1} \right]^{-1} \end{aligned} \quad (5.11)$$

Proof. Based on the definition of KLD, it can be deduced that

$$\begin{aligned} \sum_{j=0}^{M_k} \beta_j^i \mathcal{D}_{KL}(p^i \| p_j^i) &= \sum_{j=0}^{M_k} \beta_j^i \int p^i(\mathbf{x}^i) \ln \frac{p^i(\mathbf{x}^i)}{p_j^i(\mathbf{x}^i)} d\mathbf{x}^i \\ &= \int p^i(\mathbf{x}^i) \left[\sum_{j=0}^{M_k} \beta_j^i \ln \frac{p^i(\mathbf{x}^i)}{p_j^i(\mathbf{x}^i)} \right] d\mathbf{x}^i \\ &= \int p^i(\mathbf{x}^i) \ln \frac{p^i(\mathbf{x}^i)}{\prod_{j=0}^{M_k} [p_j^i(\mathbf{x}^i)]^{\beta_j^i}} d\mathbf{x}^i \\ &= \int p^i(\mathbf{x}^i) \ln \frac{p^i(\mathbf{x}^i)}{c p_*^i(\mathbf{x}^i)} d\mathbf{x}^i \\ &= \int p^i(\mathbf{x}^i) \ln \frac{p^i(\mathbf{x}^i)}{p_*^i(\mathbf{x}^i)} d\mathbf{x}^i - \ln c \int p^i(\mathbf{x}^i) d\mathbf{x}^i \\ &= \mathcal{D}_{KL}(p^i \| p_*^i) - \ln c \end{aligned} \quad (5.12)$$

where

$$p_*^i = \frac{1}{c} \prod_{j=0}^{M_k} [p_j^i(\mathbf{x}^i)]^{\beta_j^i} \quad (5.13)$$

with $c = \int \prod_{j=0}^{M_k} [p_j^i(\mathbf{x}^i)]^{\beta_j^i} d\mathbf{x}^i$ being a normalisation constant.

Note that $\mathcal{D}_{KL}(p^i \| p_*^i) \geq 0$ and the equality holds if and only if $p^i = p_*^i$. Then, the solution of the minimisation of the weighted KLD $\sum_{j=0}^{M_k} \beta_j^i \mathcal{D}_{KL}(p^i \| p_j^i)$ is $p^i = p_*^i$. Since the PDF $p_j^i(\mathbf{x}^i)$ takes Gaussian distribution $\mathcal{N}(\mathbf{x}^i; \mathbf{x}_{k|k}^{i,j}, \mathbf{P}_{k|k}^{i,j})$, it follows from the property of Gaussian products [120] that

$$[p_j^i(\mathbf{x}^i)]^{\beta_j^i} = \left[\mathcal{N}(\mathbf{x}^i; \mathbf{x}_{k|k}^{i,j}, \mathbf{P}_{k|k}^{i,j}) \right]^{\beta_j^i} \propto \mathcal{N}(\mathbf{x}^i; \mathbf{x}_{k|k}^{i,j}, \mathbf{P}_{k|k}^{i,j} / \beta_j^i) \quad (5.14)$$

and

$$\begin{aligned}
\prod_{j=0}^{M_k} p_j^i(\mathbf{x}^i) &= \prod_{j=0}^{M_k} \mathcal{N}(\mathbf{x}^i; \mathbf{x}_{k|k}^{i,j}, \mathbf{P}_{k|k}^{i,j}) \\
&\propto \exp \left\{ -\frac{1}{2} \left[(\mathbf{x}^i)^T \sum_{j=0}^{M_k} (\mathbf{P}_{k|k}^{i,j})^{-1} \mathbf{x}^i - \left(\sum_{j=0}^{M_k} (\mathbf{P}_{k|k}^{i,j})^{-1} \mathbf{x}_{k|k}^{i,j} \right)^T \mathbf{x}^i \right. \right. \\
&\quad \left. \left. - (\mathbf{x}^i)^T \sum_{j=0}^{M_k} (\mathbf{P}_{k|k}^{i,j})^{-1} \mathbf{x}_{k|k}^{i,j} \right] \right\} \\
&\propto \exp \left[-\frac{1}{2} (\mathbf{x}^i - \boldsymbol{\mu}^i)^T (\boldsymbol{\Sigma}^i)^{-1} (\mathbf{x}^i - \boldsymbol{\mu}^i) \right]
\end{aligned} \tag{5.15}$$

where

$$\begin{aligned}
\boldsymbol{\mu}^i &= \boldsymbol{\Sigma}^i \sum_{j=0}^{M_k} (\mathbf{P}_{k|k}^{i,j})^{-1} \mathbf{x}_{k|k}^{i,j} \\
\boldsymbol{\Sigma}^i &= \left(\sum_{j=0}^{M_k} (\mathbf{P}_{k|k}^{i,j})^{-1} \right)^{-1}
\end{aligned} \tag{5.16}$$

Then, we have

$$\begin{aligned}
p_*^i &= \frac{1}{c} \prod_{j=0}^{M_k} [p_j^i(\mathbf{x}^i)]^{\beta_j^i} \propto \prod_{j=0}^{M_k} \left[\mathcal{N}(\mathbf{x}^i; \mathbf{x}_{k|k}^{i,j}, \mathbf{P}_{k|k}^{i,j}) \right]^{\beta_j^i} \\
&\propto \prod_{j=0}^{M_k} \mathcal{N}(\mathbf{x}^i; \mathbf{x}_{k|k}^{i,j}, \mathbf{P}_{k|k}^{i,j} / \beta_j^i) \propto \mathcal{N}(\mathbf{x}^i; \mathbf{x}_{k|k}^i, \mathbf{P}_{k|k}^i)
\end{aligned} \tag{5.17}$$

Therefore, the optimal PDF p_*^i can be formulated as

$$p_*^i = \frac{1}{c_1} \mathcal{N}(\mathbf{x}^i; \mathbf{x}_{k|k}^i, \mathbf{P}_{k|k}^i) \tag{5.18}$$

where c_1 is a normalisation constant.

Since $\int p_*^i(\mathbf{x}^i) d\mathbf{x}^i = 1$, and $\int \mathcal{N}(\mathbf{x}^i; \mathbf{x}_{k|k}^i, \mathbf{P}_{k|k}^i) d\mathbf{x}^i = 1$, it follows from (5.18) that $c_1 = 1$ and $p_*^i = \mathcal{N}(\mathbf{x}^i; \mathbf{x}_{k|k}^i, \mathbf{P}_{k|k}^i)$. $\square$

Remark 5.2. *Proposition 5.2 reveals that the optimal solution is the weighted geometric mean of all PDFs, which coincides with the generalised CI in information fusion [27, 121]. This means that the proposed algorithm holds similar property as generalised CI. That is, the proposed method is robust against the unknown corre-*

Table 5.1: Computational complexity comparison.

Algorithm	Computation term	Computational complexity
Moment-preserving (5.1)	Computing $\mathbf{x}_{k k}^i$	$\mathcal{O}((M_k + 1)n)$
	Computing $\mathbf{P}_{k k}^i$	$\mathcal{O}((M_k + 1)(3n^2 + n))$
Proposed approach (5.11)	Computing $\mathbf{x}_{k k}^i$	$\mathcal{O}((M_k + 1)(n^3 + 2n^2) + n^2)$
	Computing $\mathbf{P}_{k k}^i$	$\mathcal{O}((M_k + 1)(n^3 + n^2) + n^3)$

lations among the information sources. Therefore, one can safely predict that the proposed JPDA exhibits improved performance when targets are moving closely.

Remark 5.3. By using the computational complexity of the related matrix manipulations [122], the computational complexity comparisons of the proposed Gaussian mixture approximation and the simple moment-preserving approach of the original JPDA are presented in Table 5.1. From this table, one can note that the proposed approach has higher computational complexity than the simple moment-matching method due to the inversion operation of $\mathbf{P}_{k|k}^{i,j}$. However, the computational complexities of both algorithms are scalable with respect to the number of measurements.

5.2.2 Implementation of Information-Theoretic JPDA

Similar to the standard JPDA, the proposed information-theoretic JPDA algorithm also forms tracks out of the marginal measurement-to-track association distributions, and thus is a track-oriented approach. The implementation of this new algorithm is the same as the Gibbs-JPDA, discussed in Chapter 4, except for the mixture approximation step. The complete pseudo code of the proposed information-theoretic JPDA is summarised in Algorithm 5.1.

Algorithm 5.1 Information-Theoretic Joint Probabilistic Data Association Filter

- 1: Predict the existing targets using target dynamics
 - 2: Update each track with all possible measurements by Kalman filter
 - 3: Calculate the marginal probability $p(\theta_k^i | \mathcal{Z}_k)$
 - 4: Calculate the posterior existence probability $p(\chi_k^i | \mathcal{Z}_k)$
 - 5: Calculate the existence-conditioned marginal probability $p(\theta_k^i | \chi_k^i, \mathcal{Z}_k)$
 - 6: Use the results shown in Proposition 5.2 to update N_{k-1} existing targets
 - 7: Generate M_k new targets (one for each measurement)
 - 8: Perform track confirmation and deletion based using the existence probability $p(\chi_k^i | \mathcal{Z}_k)$
-

5.3 Performance Analysis

In this subsection, the performance of the proposed information-driven mixture approximation is analysed. Intuitively, continuous miss detection results in significant information loss, leading to the divergence of the estimation filter. Therefore, the performance is analysed under the condition $P_D = 1$ in this subsection, while the effect of P_D on the estimation performance will be empirically studied in the simulation parts. We first analyse the mean performance of the proposed estimator and then derive a lower bound and an upper bound of the error covariance.

Theorem 5.1. (*Mean Performance*) *Under the condition $P_D = 1$, the proposed Gaussian mixture approximation, given by (5.11), yields unbiased estimation if every hypothesis leads an unbiased estimate, i.e., $\mathbb{E} \left[x_{k|k}^{i,j} \right] = x_k^i$.*

Proof. Define $\mathbf{Y}_{k|k}^i = \left(\mathbf{P}_{k|k}^i \right)^{-1}$ as the information matrix and $\mathbf{y}_{k|k}^i = \left(\mathbf{P}_{k|k}^i \right)^{-1} \mathbf{x}_{k|k}^i$ as the information contribution. Then, Eq. (5.11) can be rewritten as

$$\mathbf{y}_{k|k}^i = \sum_{j=0}^{M_k} \beta_j^i \mathbf{y}_{k|k}^{i,j}, \quad \mathbf{Y}_{k|k}^i = \sum_{j=0}^{M_k} \beta_j^i \mathbf{Y}_{k|k}^{i,j} \quad (5.19)$$

which reveals that the proposed mixture reduction algorithm can be viewed as a weighted sum of information terms.

The remaining proof is given by mathematical induction. Consider the case that the i th target has two valid measurements, e.g., $j = 1, 2$. Then, (5.19) reduces to

$$\begin{aligned} \mathbf{y}_{k|k}^i &= \beta_1^i \mathbf{y}_{k|k}^{i,1} + (1 - \beta_1^i) \mathbf{y}_{k|k}^{i,2} \\ \mathbf{Y}_{k|k}^i &= \beta_1^i \mathbf{Y}_{k|k}^{i,1} + (1 - \beta_1^i) \mathbf{Y}_{k|k}^{i,2} \end{aligned} \quad (5.20)$$

Applying the matrix inversion lemma, we have

$$\begin{aligned} \left(\mathbf{Y}_{k|k}^i \right)^{-1} \left(\beta_1^i \mathbf{Y}_{k|k}^{i,1} \right) &= \left[\beta_1^i \mathbf{Y}_{k|k}^{i,1} + (1 - \beta_1^i) \mathbf{Y}_{k|k}^{i,2} \right]^{-1} \left(\beta_1^i \mathbf{Y}_{k|k}^{i,1} \right) \\ &= \left\{ \left(\beta_1^i \mathbf{Y}_{k|k}^{i,1} \right)^{-1} - \left(\beta_1^i \mathbf{Y}_{k|k}^{i,1} \right)^{-1} \mathbf{A}^{-1} \left(\beta_1^i \mathbf{Y}_{k|k}^{i,1} \right)^{-1} \right\} \left(\beta_1^i \mathbf{Y}_{k|k}^{i,1} \right) \\ &= \mathbf{I}_n - \left(\beta_1^i \mathbf{Y}_{k|k}^{i,1} \right)^{-1} \mathbf{A}^{-1} \\ &= \mathbf{I}_n - \mathbf{B} \end{aligned} \quad (5.21)$$

where

$$\begin{aligned}\mathbf{A} &\triangleq \left(\beta_1^i \mathbf{Y}_{k|k}^{i,1}\right)^{-1} + \left[(1 - \beta_1^i) \mathbf{Y}_{k|k}^{i,2}\right]^{-1} \\ \mathbf{B} &\triangleq \left(\beta_1^i \mathbf{Y}_{k|k}^{i,1}\right)^{-1} \mathbf{A}^{-1}\end{aligned}\quad (5.22)$$

Note that

$$\begin{aligned}(1 - \beta_1^i) \mathbf{Y}_{k|k}^{i,2} \left(\beta_1^i \mathbf{Y}_{k|k}^{i,1}\right)^{-1} + \mathbf{I}_n &= (1 - \beta_1^i) \mathbf{Y}_{k|k}^{i,2} \left(\beta_1^i \mathbf{Y}_{k|k}^{i,1}\right)^{-1} \\ &\quad + (1 - \beta_1^i) \mathbf{Y}_{k|k}^{i,2} \frac{\left(\mathbf{Y}_{k|k}^{i,2}\right)^{-1}}{1 - \beta_1^i} \\ &= (1 - \beta_1^i) \mathbf{Y}_{k|k}^{i,2} \mathbf{A}\end{aligned}\quad (5.23)$$

$$\begin{aligned}\mathbf{I}_n + (1 - \beta_1^i) \mathbf{Y}_{k|k}^{i,2} \left(\beta_1^i \mathbf{Y}_{k|k}^{i,1}\right)^{-1} &= \left(\beta_1^i \mathbf{Y}_{k|k}^{i,1}\right) \left(\beta_1^i \mathbf{Y}_{k|k}^{i,1}\right)^{-1} \\ &\quad + (1 - \beta_1^i) \mathbf{Y}_{k|k}^{i,2} \left(\beta_1^i \mathbf{Y}_{k|k}^{i,1}\right)^{-1} \\ &= \mathbf{Y}_{k|k}^i \left(\beta_1^i \mathbf{Y}_{k|k}^{i,1}\right)^{-1}\end{aligned}\quad (5.24)$$

Combining (5.23) and (5.24) yields

$$\left(\mathbf{Y}_{k|k}^i\right)^{-1} (1 - \beta_1^i) \mathbf{Y}_{k|k}^{i,2} = \left(\beta_1^i \mathbf{Y}_{k|k}^{i,1}\right)^{-1} \mathbf{A}^{-1} = \mathbf{B} \quad (5.25)$$

Based on (5.21) and (5.25), it is straightforward to evaluate the expectation $\mathbb{E} \left[\mathbf{x}_{k|k}^i \right]$ as

$$\begin{aligned}\mathbb{E} \left[\mathbf{x}_{k|k}^i \right] &= \mathbb{E} \left[\left(\mathbf{Y}_{k|k}^i\right)^{-1} \mathbf{y}_{k|k}^i \right] \\ &= \mathbb{E} \left[\left(\mathbf{Y}_{k|k}^i\right)^{-1} \left(\beta_1^i \mathbf{Y}_{k|k}^{i,1} \mathbf{x}_{k|k}^{i,1} + (1 - \beta_1^i) \mathbf{Y}_{k|k}^{i,2} x_{k|k}^{i,2} \right) \right] \\ &= \mathbb{E} \left[(\mathbf{I}_n - \mathbf{B}) \mathbf{x}_{k|k}^{i,1} + \mathbf{B} x_{k|k}^{i,2} \right] \\ &= (\mathbf{I}_n - \mathbf{B}) \mathbb{E} \left[\mathbf{x}_{k|k}^{i,1} \right] + \mathbf{B} \mathbb{E} \left[x_{k|k}^{i,2} \right] \\ &= \mathbf{x}_k^i\end{aligned}\quad (5.26)$$

since $\mathbb{E} \left[\mathbf{x}_{k|k}^{i,1} \right] = \mathbf{x}_k^i$ and $\mathbb{E} \left[x_{k|k}^{i,2} \right] = x_k^i$ according to the property of Kalman filter.

At the inductive step, assume that the i th target has $M_k - 1$ valid measurements, e.g., $j = 1, 2, \dots, M_k - 1$ and the proposed Gaussian approximation approach yields

unbiased estimation

$$\mathbf{y}_{k|k}^i = \sum_{j=1}^{M_k-1} \beta_j^i \mathbf{y}_{k|k}^{i,j}, \quad \mathbf{Y}_{k|k}^i = \sum_{j=1}^{M_k-1} \beta_j^i \mathbf{Y}_{k|k}^{i,j} \quad (5.27)$$

Following similar procedures, shown in (5.20)-(5.26), it is easy to verify that, if the i th target has M_k valid measurements, e.g., $j = 1, 2, \dots, M_k$, the proposed Gaussian approximation approach yields unbiased estimation. Then, the proof is completed by mathematical induction. $\square$

Remark 5.4. Notice that the expectation $\mathbb{E} \left[x_{k|k}^i \right]$ is evaluated based on the posterior $p(x_k^i | \chi_k^i, Z_k)$, which is a function of x_k^i given current measurements Z_k . Since data association only depends on the target dynamics model and current measurements Z_k , the marginal probability β_j^i is determined once the association process is finished and therefore β_j^i is considered as a known constant in the analysis of Theorem 5.1.

Remark 5.5. The establishment of Theorem 5.1 requires the assumption that every hypothesis leads an unbiased estimate. Notice that each single association hypothesis is updated by classical linear Kalman filter, which provides unbiased estimate provided that the noises are Gaussian. For this reason, we utilise the unbiased assumption of each hypothesis in the analysis.

Theorem 5.1 reveals that, if the original estimator for single association pair update is unbiased, the proposed algorithm yields unbiased estimation. Now, let us investigate the error covariance performance, which quantifies the estimation accuracy, of the propose information-theoretic approach. To derive the bound on the covariance matrix $\mathbf{P}_{k|k}^i$, we consider the following linear system for the i th target

$$\begin{aligned} \mathbf{x}_k^i &= \mathbf{F}_{k-1}^i \mathbf{x}_{k-1}^i + \mathbf{w}_{k-1}^i \\ \mathbf{z}_k^i &= \mathbf{H}_k^i \mathbf{x}_k^i + \mathbf{v}_k^i \end{aligned} \quad (5.28)$$

where $\mathbf{F}_{k-1}^i \in \mathbb{R}^{n \times n}$ and $\mathbf{H}_k^i \in \mathbb{R}^{m \times n}$ correspond to the system matrix and observation matrix, respectively. The signals $\mathbf{w}_k^i \in \mathbb{R}^n$ and $\mathbf{v}_k^i \in \mathbb{R}^m$ are process noise and measurement noise, which are assumed to be zero-mean Gaussian process with covariances $\mathbf{Q}_k^i \in \mathbb{R}^{n \times n}$ and $\mathbf{R}_k^i \in \mathbb{R}^{m \times m}$, respectively.

Theorem 5.2. (*Error Covariance Performance*) Suppose $P_D = 1$ and the linear system (5.28) is uniformly controllable and uniformly observable. If the initial error covariance $\mathbf{P}_{0|0}^{i,j} > 0$, then the estimation error given by (5.11) is asymptotically bounded in a mean square sense and the corresponding error covariance is uniformly

bounded from below and from above as

$$\underline{\mathbf{P}} \leq \mathbf{P}_{k|k}^i \leq \overline{\mathbf{P}} \quad (5.29)$$

where $\underline{\mathbf{P}}$ and $\overline{\mathbf{P}}$ are positive matrices.

Proof. According to JPDA, the error covariance update of each association pair $\theta_i = j$ is obtained by standard Kalman filter as

$$\begin{aligned} \left(\mathbf{P}_{k|k}^{i,j}\right)^{-1} &= \left(\mathbf{P}_{k|k-1}^{i,j}\right)^{-1} + \left(\mathbf{H}_k^i\right)^T \mathbf{R}_k^{-1} \mathbf{H}_k^i \\ &= \left(\mathbf{F}_{k-1}^i \mathbf{P}_{k-1|k-1}^i \left(\mathbf{F}_{k-1}^i\right)^T + \mathbf{Q}_{k-1}^i\right)^{-1} + \left(\mathbf{H}_k^i\right)^T \mathbf{R}_k^{-1} \mathbf{H}_k^i \\ &= \left(\mathbf{F}_{k-1}^i \left[\sum_{j=1}^{M_k} \beta_j^i \left(\mathbf{P}_{k-1|k-1}^{i,j}\right)^{-1}\right]^{-1} \left(\mathbf{F}_{k-1}^i\right)^T + \mathbf{Q}_{k-1}^i\right)^{-1} + \left(\mathbf{H}_k^i\right)^T \mathbf{R}_k^{-1} \mathbf{H}_k^i \end{aligned} \quad (5.30)$$

Note that the previous equation is derived based on the information form of Kalman filter. According to Lemma C.1 in Appendix C, the hypothesis-conditioned error covariance is uniformly bounded from below and from above as

$$\underline{\mathbf{P}}_i \leq \mathbf{P}_{k|k}^{i,j} \leq \overline{\mathbf{P}}_i \quad (5.31)$$

where $\underline{\mathbf{P}}_i$ and $\overline{\mathbf{P}}_i$ are positive matrices.

Based on (5.31), we can choose matrices $\mathbf{P}_{k|k}^U$ and $\mathbf{P}_{k|k}^L$ at each scan as

$$\begin{aligned} \mathbf{P}_{k|k}^{i,U} &\triangleq \left\{ \mathbf{P}_{k|k}^{i,j_{\max}} \mid \mathbf{P}_{k|k}^{i,j_{\max}} - \mathbf{P}_{k|k}^{i,j} \geq 0, j \in [M_k] \right\} \\ \mathbf{P}_{k|k}^{i,L} &\triangleq \left\{ \mathbf{P}_{k|k}^{i,j_{\min}} \mid \mathbf{P}_{k|k}^{i,j_{\min}} - \mathbf{P}_{k|k}^{i,j} \leq 0, j \in [M_k] \right\} \end{aligned} \quad (5.32)$$

where $[M_k] = \{1, 2, \dots, M_k\}$.

Then, one can imply that

$$\begin{aligned} \sum_{j=1}^{M_k} \beta_j^i \left(\mathbf{P}_{k|k}^{i,j}\right)^{-1} &\leq \sum_{j=1}^{M_k} \beta_j^i \left(\mathbf{P}_{k|k}^{i,L}\right)^{-1} = \left(\mathbf{P}_{k|k}^{i,L}\right)^{-1} \\ \sum_{j=1}^{M_k} \beta_j^i \left(\mathbf{P}_{k|k}^{i,j}\right)^{-1} &\geq \sum_{j=1}^{M_k} \beta_j^i \left(\mathbf{P}_{k|k}^{i,U}\right)^{-1} = \left(\mathbf{P}_{k|k}^{i,U}\right)^{-1} \end{aligned} \quad (5.33)$$

Substituting (5.33) into (5.30) yields

$$\begin{aligned} \left(\mathbf{P}_{k|k}^{i,j}\right)^{-1} &\leq \left(\mathbf{F}_{k-1}^i \mathbf{P}_{k|k}^{i,L} \left(\mathbf{F}_{k-1}^i\right)^T + \mathbf{Q}_{k-1}^i\right)^{-1} + \left(\mathbf{H}_k^i\right)^T \mathbf{R}_k^{-1} \mathbf{H}_k^i \\ \left(\mathbf{P}_{k|k}^{i,j}\right)^{-1} &\geq \left(\mathbf{F}_{k-1}^i \mathbf{P}_{k|k}^{i,U} \left(\mathbf{F}_{k-1}^i\right)^T + \mathbf{Q}_{k-1}^i\right)^{-1} + \left(\mathbf{H}_k^i\right)^T \mathbf{R}_k^{-1} \mathbf{H}_k^i \end{aligned} \quad (5.34)$$

Note that the right hand sides of (5.34) can be considered as two standard Kalman filter error covariance updates as

$$\begin{aligned} \mathbf{P}_{k|k}^{i,j} &\geq \Psi_k \left(\Psi_{k-1} \cdots \left(\Psi_1 \left(\mathbf{P}_{0|0}^{i,L} \right) \cdots \right) \right) \\ \mathbf{P}_{k|k}^{i,j} &\leq \Psi_k \left(\Psi_{k-1} \cdots \left(\Psi_1 \left(\mathbf{P}_{0|0}^{i,U} \right) \cdots \right) \right) \end{aligned} \quad (5.35)$$

where the operator $\Psi_k(\cdot)$ is defined as

$$\Psi_k(\mathbf{P}) \triangleq \left[\left(\mathbf{F}_{k-1}^i \mathbf{P} \left(\mathbf{F}_{k-1}^i \right)^T + \mathbf{Q}_{k-1}^i \right)^{-1} + \left(\mathbf{H}_k^i \right)^T \mathbf{R}_k^{-1} \mathbf{H}_k^i \right]^{-1} \quad (5.36)$$

Consequently, the estimation error given by (5.11) is asymptotically bounded in a mean square sense and the corresponding error covariance is uniformly bounded from below and from above by using Lemma C.1. $\square$

5.4 Simulation Studies

In this section, the effectiveness of the proposed MTT algorithm is demonstrated through extensive empirical numerical simulations.

5.4.1 Simulation Setup

Our experiments explore a very challenging scenario, involving ten closely moving targets. These ten targets appear at time step from $t = 0s$ to $t = 90s$ at every 10s interval. The state vector contains planar position and velocity. We assume that all targets share the same motion model and use the well-known CV model for target prediction. The CV model is defined as

$$\mathbf{x}_k^i = \mathbf{F}_{CV} \mathbf{x}_{k-1}^i + \mathbf{G} w_{k-1}^i \quad (5.37)$$

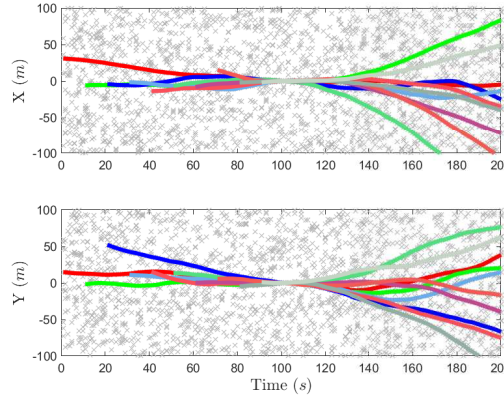


Figure 5.3: Sample ground truth and measurements (grey 'x' for measurements and colour lines for targets).

with

$$\mathbf{F}_{CV} \triangleq \mathbf{I}_2 \otimes \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}, \quad \mathbf{G} \triangleq \begin{bmatrix} T_s^2/2 \\ T_s \\ T_s^2/2 \\ T_s \end{bmatrix} \quad (5.38)$$

where $T_s = 1s$ is the sampling period, and $w_k^i \sim \mathcal{N}(\cdot; 0, \sigma_v^2)$ the Gaussian process noise with $\sigma_v = 0.1m/s^2$.

The linear position measurements, generated with $P_D = 0.9$, are modelled by

$$\mathbf{z}_k^i = \begin{bmatrix} p_{x,k}^i \\ p_{y,k}^i \end{bmatrix} + \mathbf{v}_k^i \quad (5.39)$$

where $(p_{x,k}^i, p_{y,k}^i)$ denotes the 2D inertial position of the i th target, $\mathbf{v}_k^i \sim \mathcal{N}(\cdot; 0, \mathbf{R}_k^i)$ the Gaussian measurement noise with $\mathbf{R}_k^i = \sigma_z^2 \text{diag}([1, 1])$. The clutter is assumed to be uniformly distributed in the surveillance region with its number being Poisson with 10 average returns at each scan. Snapshots of one sample trajectory for each target and 'transformed' measurements are depicted in Fig. 5.3.

In order to cover the entire region of interest, the target birth model is $\lambda_B(\mathbf{x}) = \lambda_B \mathcal{N}(\mathbf{x}; \mathbf{x}_b, \mathbf{P}_b)$ with $\lambda_B = 0.05$, $\mathbf{x}_b = [0, 0, 0, 0]^T$, $\mathbf{P}_b = \text{diag}([100^2, 1, 100^2, 1])$. A tentative track is confirmed if the existence probability satisfies $p(\chi_k^i | \mathcal{Z}_k) \geq 0.8$ and a confirmed track is deleted immediately once $p(\chi_k^i | \mathcal{Z}_k) \leq 0.1$. The OSPA distance with $p = 1$, $c = 10$ is utilised in this chapter to evaluate the performance of the proposed algorithm.

5.4.2 Performance Evaluation

We compare the proposed JPDA filter with standard JPDA and measurement-driven JPDA [59] for these three different cases. Note that the measure-driven JPDA can avoid the multi-modality issue but cannot provide the information on target label. In order to make fair comparisons, the Gibbs sampling-aided marginalisation is also utilised in other two algorithms.

The results of mean OSPA distance obtained by 100 Monte-Carlo runs are shown in Fig. 5.4. The peaks of mean OSPA distance before 100s in this figure are resulted from track confirmation for target birth. Since all targets cross each other at around 100s, all filters cannot correctly assign measurements to targets, which in turn results in performance degradation. Although the measurement-driven JPDA can also mitigate the multi-modality issue in Gaussian mixture approximation, the resultant solution is sensitive to measurement noise, as confirmed by Fig. 5.4. It is clear from Fig. 5.4 that the proposed information-theoretic JPDA outperforms other two algorithms in terms of tracking accuracy and shows strong robustness against the variation of measurement noise in all cases. The reason is that minimising the proposed KLD provides approximate mode selection in Gaussian mixture reduction for updating JPDA and therefore can alleviate the multi-modality issue when targets are moving closely around 100s. Therefore, the proposed information-theoretic approach is a competitive alternative to the original JPDA and can be expected to address the Gaussian mixture approximation problem in related methods.

5.5 Conclusions

This chapter proposes a new information-theoretic JPDA filter for tracking unknown number of closely-moving targets in a cluttered environment. The proposed JPDA filter is derived through the minimisation of a weighted reverse KLD to approximate the posterior Gaussian mixture PDF. The resultant approximation approach turns out to show similar structure as the generalised covariance intersection in sensor fusion. Theoretical analysis of mean performance and error covariance performance with ideal detection probability is presented to provide better insights of the proposed approach. Extensive empirical simulations are undertaken to validate the performance of the proposed MTT algorithm.

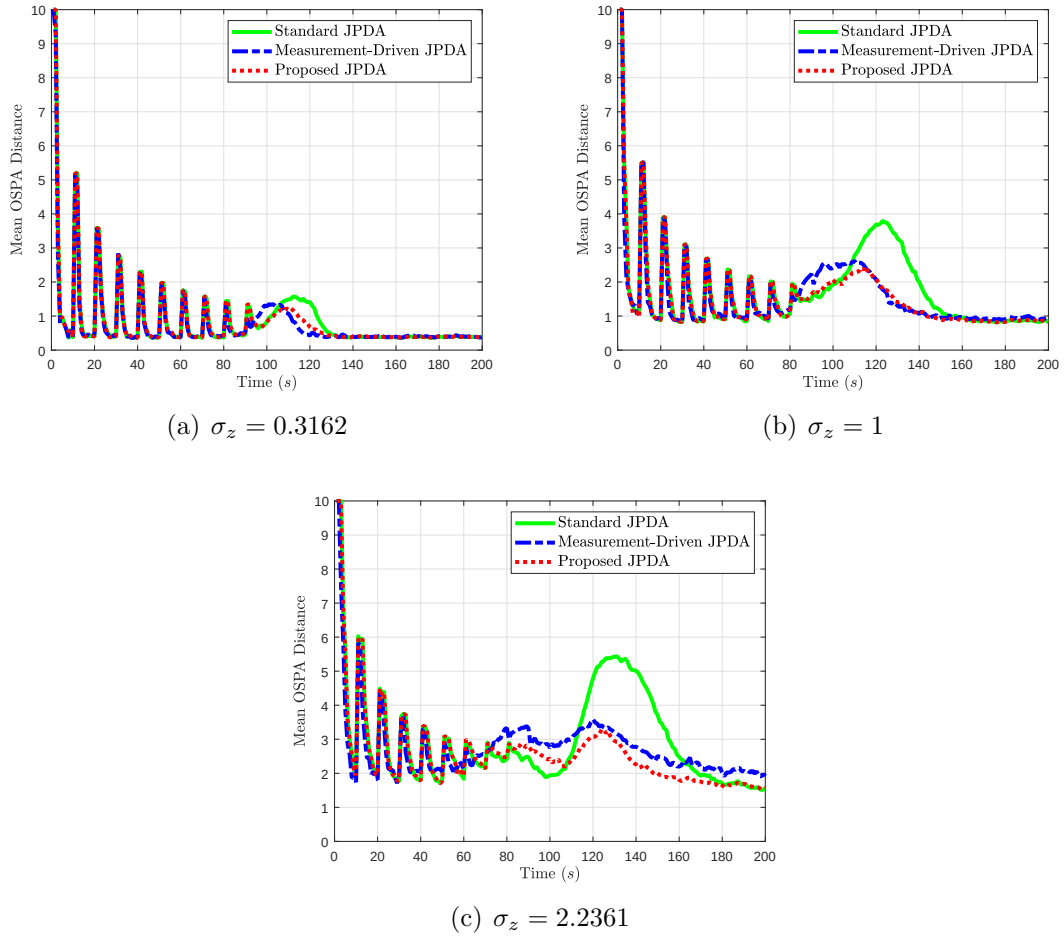


Figure 5.4: Mean OSPA distance of 100 Monte-Carlo simulations.

Chapter 6

Distributed Joint Probabilistic Data Association Filter with Hybrid Fusion

6.1 Introduction

Generally, a distributed MTT algorithm contains two main components: local multi-target tracker for each sensor node and distributed information fusion algorithm for locally connected sensors. The previous two chapters address the problem of local MTT. In the fusion stage, each sensor node communicates with its neighbours through a network topology to perform track-to-track association to find the source origin of local estimates that come from different sensors. After discriminating the origins of the tracks with different sensor labels, proper fusion strategies can be exploited to fuse local estimates. Note that although distributed multi-sensor target tracking for single target is well-established, its direct extension to the MTT scenario is not straightforward due to the measurement origin uncertainty and hence requires careful adjustment. This is the main concern of this chapter.

This chapter develops distributed MTT algorithms by exploiting different fusion strategies: distributed JPDA with consensus on measurement (DJPDA-CM), distributed JPDA with consensus on information (DJPDA-CI) and distributed JPDA with hybrid consensus (DJPDA-HC). The main contributions are:

- The conventional CM and CI strategies are extended to the MTT scenario,

addressing the inherent data association uncertainty issue. Specifically, except for local estimates, the data association uncertainty-related term of each sensor node is also shared with its neighbours for information fusion.

- A new distributed multi-target tracking algorithm with hybrid fusion strategy, termed as DJPDA-HC, is proposed by exploiting the benefits of both DJPDA-CM and DJPDA-CI. The proposed DJPDA-HC filter is a fully distributed tracker without utilising any global information; thus, this approach has strong robustness against sensor failures. Theoretical analysis and extensive simulations demonstrate that DJPDA-HC outperforms DJPDA-CM and DJPDA-CI under various conditions.

6.2 Centralised JPDA Filter: A Benchmark

An optimal fusion strategy and benchmark for performance evaluation of distributed state estimation algorithms is centralised estimation, which processes the measurements from all sensors simultaneously through a fusion centre. This section will derive the centralised JPDA, which will be utilised as a benchmark for the performance evaluation of the algorithms developed. For centralised implementation of multi-sensor JPDA filter, we will utilise the information filter, which is proved to be useful in information fusion [123].

6.2.1 Information-Form JPDA

This chapter considers the following linear system

$$\begin{aligned}\mathbf{x}_k^i &= \mathbf{F}_{k-1}^i \mathbf{x}_{k-1}^i + \mathbf{w}_{k-1}^i \\ \mathbf{z}_k^i &= \mathbf{H}_k^i \mathbf{x}_k^i + \mathbf{v}_k^i\end{aligned}\tag{6.1}$$

where $\mathbf{F}_{k-1}^i \in \mathbb{R}^{n \times n}$ and $\mathbf{H}_k^i \in \mathbb{R}^{m \times n}$ correspond to the system matrix and observation matrix, respectively. The signals $\mathbf{w}_k^i \in \mathbb{R}^n$ and $\mathbf{v}_k^i \in \mathbb{R}^m$ are process noise and measurement noise, which are assumed to be zero-mean Gaussian process with covariances $\mathbf{Q}_k^i \in \mathbb{R}^{n \times n}$ and $\mathbf{R}_k^i \in \mathbb{R}^{m \times m}$, respectively.

JPDA assumes that each measurement can originate from a number of candidate targets in data association. Therefore, the posterior probability distribution of each target obtained from JPDA filter is a Gaussian mixture distribution. Since propaga-

tion of the Gaussian mixture distribution over time is practically intractable due to the explosion of mixture components, JPDA utilises a single Gaussian distribution to approximate the Gaussian mixture at each time instant to reduce the computational burden. This can be achieved by updating the state estimation based on a pseudo innovation term, i.e., a weighted sum of the original innovation terms¹ [51, 124],

$$\tilde{\mathbf{z}}_k^i = \sum_{j=1}^{M_k} \beta_j^i \tilde{\mathbf{z}}_{j,k}^i = \sum_{j=1}^{M_k} \beta_j^i (\mathbf{z}_{j,k} - \mathbf{H}_k^i \mathbf{x}_{k|k-1}^i) \quad (6.2)$$

where β_j^i represents the marginal association probability that the j th measurement is associated with the i th target. Details of how to calculate the marginal association probability β_j^i can be found in Chapter 4.

The original JPDA paradigm can be obtained by applying the pseudo innovation term $\tilde{\mathbf{z}}_k^i$ to standard Kalman filter as

• **Prediction:**

$$\begin{aligned} \mathbf{x}_{k|k-1}^i &= \mathbf{F}_{k-1}^i \mathbf{x}_{k-1|k-1}^i \\ \mathbf{P}_{k|k-1}^i &= \mathbf{F}_{k-1}^i \mathbf{P}_{k-1|k-1}^i (\mathbf{F}_{k-1}^i)^T + \mathbf{Q}_{k-1}^i \end{aligned} \quad (6.3)$$

• **Update:**

$$\begin{aligned} \mathbf{S}_k^i &= \mathbf{H}_k^i \mathbf{P}_{k|k-1}^i (\mathbf{H}_k^i)^T + \mathbf{R}_k^i \\ \mathbf{K}_k^i &= \mathbf{P}_{k|k-1}^i (\mathbf{H}_k^i)^T (\mathbf{S}_k^i)^{-1} \\ \mathbf{x}_{k|k}^i &= \mathbf{x}_{k|k-1}^i + \mathbf{K}_k^i \tilde{\mathbf{z}}_k^i \\ \mathbf{P}_{k|k}^i &= \mathbf{P}_{k|k-1}^i - \mathbf{K}_k^i (1 - \beta_0^i) \mathbf{S}_k^i (\mathbf{K}_k^i)^T + \mathbf{K}_k^i \bar{\mathbf{P}}_k (\mathbf{K}_k^i)^T \end{aligned} \quad (6.4)$$

where $\bar{\mathbf{P}}_k$ is a positive semi-definite matrix representing the measurement origin uncertainty and takes the form

$$\bar{\mathbf{P}}_k = \sum_{j=1}^{M_k} \beta_j^i \tilde{\mathbf{z}}_{j,k}^i (\tilde{\mathbf{z}}_{j,k}^i)^T - \tilde{\mathbf{z}}_k^i (\tilde{\mathbf{z}}_k^i)^T \quad (6.5)$$

For the purpose of deriving information form JPDA filter, define $\mathbf{Y}_{k|k}^i = (\mathbf{P}_{k|k}^i)^{-1}$ and $\mathbf{y}_{k|k}^i = (\mathbf{P}_{k|k}^i)^{-1} \mathbf{x}_{k|k}^i$ as the information matrix and the information vector, respectively. Denote $\boldsymbol{\Omega}_k^i = (\mathbf{H}_k^i)^T (\mathbf{R}_k^i)^{-1} \mathbf{H}_k^i$, $\boldsymbol{\delta}_k^i = (\mathbf{H}_k^i)^T (\mathbf{R}_k^i)^{-1} \sum_{j=1}^{M_k} \beta_j^i \mathbf{z}_{j,k}^i$. Using

¹This is an equivalent form of using moment matching, shown in Eq. (5.1), in JPDA.

matrix inversion lemma, the gain of JPDA filter $\mathbf{K}_k^i$ can be obtained as

$$\begin{aligned}
\mathbf{K}_k^i &= \mathbf{P}_{k|k-1}^i (\mathbf{H}_k^i)^T \left(\mathbf{H}_k^i \mathbf{P}_{k|k-1}^i (\mathbf{H}_k^i)^T + \mathbf{R}_k^i \right)^{-1} \\
&= \mathbf{P}_{k|k-1}^i (\mathbf{H}_k^i)^T \left[(\mathbf{R}_k^i)^{-1} - (\mathbf{R}_k^i)^{-1} \mathbf{H}_k^i \right. \\
&\quad \times \left. \left[(\mathbf{P}_{k|k-1}^i)^{-1} + (\mathbf{H}_k^i)^T (\mathbf{R}_k^i)^{-1} \mathbf{H}_k^i \right]^{-1} (\mathbf{H}_k^i)^T (\mathbf{R}_k^i)^{-1} \right] \\
&= \left[(\mathbf{P}_{k|k-1}^i)^{-1} + (\mathbf{H}_k^i)^T (\mathbf{R}_k^i)^{-1} \mathbf{H}_k^i \right]^{-1} (\mathbf{H}_k^i)^T (\mathbf{R}_k^i)^{-1}
\end{aligned} \tag{6.6}$$

Substituting Eq. (6.6) in Eq. (6.4) gives the state estimation as

$$\begin{aligned}
\mathbf{x}_{k|k}^i &= \mathbf{x}_{k|k-1}^i + \left[(\mathbf{P}_{k|k-1}^i)^{-1} + (\mathbf{H}_k^i)^T (\mathbf{R}_k^i)^{-1} \mathbf{H}_k^i \right]^{-1} (\mathbf{H}_k^i)^T (\mathbf{R}_k^i)^{-1} \tilde{\mathbf{z}}_k^i \\
&= \mathbf{x}_{k|k-1}^i + \left[(\mathbf{P}_{k|k-1}^i)^{-1} + (\mathbf{H}_k^i)^T (\mathbf{R}_k^i)^{-1} \mathbf{H}_k^i \right]^{-1} \\
&\quad \times (\mathbf{H}_k^i)^T (\mathbf{R}_k^i)^{-1} \left(\sum_{j=1}^{M_k} \beta_j^i \mathbf{z}_k^j - (1 - \beta_0^i) \mathbf{H}_k^i \mathbf{x}_{k|k-1}^i \right) \\
&= \mathbf{x}_{k|k-1}^i + (\mathbf{Y}_{k|k-1}^i + \mathbf{\Omega}_k^i)^{-1} [\boldsymbol{\delta}_k^i - (1 - \beta_0^i) \mathbf{\Omega}_k^i \mathbf{x}_{k|k-1}^i] \\
&= (\mathbf{Y}_{k|k-1}^i + \mathbf{\Omega}_k^i)^{-1} (\mathbf{y}_{k|k-1}^i + \boldsymbol{\delta}_k^i + \beta_0^i \mathbf{\Omega}_k^i \mathbf{x}_{k|k-1}^i)
\end{aligned} \tag{6.7}$$

Based on the matrix inversion lemma, the update of the information matrix $\mathbf{Y}_{k|k-1}^i$ is derived as

$$\begin{aligned}
\mathbf{Y}_{k|k}^i &= \left\{ \mathbf{P}_{k|k-1}^i - \mathbf{K}_k^i (1 - \beta_0^i) (\mathbf{K}_k^i)^T + \mathbf{K}_k^i \bar{\mathbf{P}}_k^i (\mathbf{K}_k^i)^T \right\}^{-1} \\
&= \left\{ \mathbf{P}_{k|k-1}^i - \mathbf{K}_k^i [(1 - \beta_0^i) \mathbf{S}_k^i - \bar{\mathbf{P}}_k^i] (\mathbf{K}_k^i)^T \right\}^{-1} \\
&= \mathbf{Y}_{k|k-1}^i + \bar{\mathbf{\Omega}}_k^i
\end{aligned} \tag{6.8}$$

where the information matrix contribution $\bar{\mathbf{\Omega}}_k^i$ is given by

$$\bar{\mathbf{\Omega}}_k^i = \mathbf{Y}_{k|k-1}^i \mathbf{K}_k^i \left\{ [(1 - \beta_0^i) \mathbf{S}_k^i - \bar{\mathbf{P}}_k^i]^{-1} - (\mathbf{K}_k^i)^T \mathbf{Y}_{k|k-1}^i \mathbf{K}_k^i \right\} (\mathbf{K}_k^i)^T \mathbf{Y}_{k|k-1}^i \tag{6.9}$$

Based Eqs. (6.7) and (6.8) the information form of JPDA filter is given by

$$\begin{aligned}
\mathbf{Y}_{k|k}^i &= \mathbf{Y}_{k|k-1}^i + \bar{\mathbf{\Omega}}_k^i \\
\mathbf{x}_{k|k}^i &= (\mathbf{Y}_{k|k-1}^i + \mathbf{\Omega}_k^i)^{-1} (\mathbf{y}_{k|k-1}^i + \boldsymbol{\delta}_k^i + \beta_0^i \mathbf{\Omega}_k^i \mathbf{x}_{k|k-1}^i)
\end{aligned} \tag{6.10}$$

Note that $\boldsymbol{\delta}_k^i$ is the measurement information, which clearly reveals that JPDA utilises a weighted sum of all candidate measurements. The term $\beta_0^i \mathbf{\Omega}_k^i \mathbf{x}_{k|k-1}^i$ quantifies target miss detection. If the i th target is miss detected, then, $\beta_0^i = 1$ and

$\beta_j^i = 0$, which means $\delta_k^i = 0$. Therefore, only prior information can be utilised for update. The summation of δ_k^i and $\beta_0^i \mathbf{\Omega}_k^i \mathbf{x}_{k|k-1}^i$ determines the total amount of information that JPDA can leverage to update the i th target. Due to data association uncertainty, there exist two information matrix contributions, e.g., $\mathbf{\Omega}_k^i$ and $\bar{\mathbf{\Omega}}_k^i$ in information JPDA filter. Notice that the inverse of $\mathbf{Y}_{k|k-1}^i + \bar{\mathbf{\Omega}}_k^i$ is the posterior covariance matrix, including data association uncertainty, of the i th target, provided by JPDA filter, whereas the inverse of $\mathbf{Y}_{k|k-1}^i + \mathbf{\Omega}_k^i$ can be interpreted as the ideal covariance matrix of the i th target without any association uncertainty. It is easy to verify that $\bar{\mathbf{\Omega}}_k^i = \mathbf{\Omega}_k^i$ when $\beta_0^i = 0$ and $\bar{\mathbf{P}}_k^i = 0$, e.g., no data association uncertainty. This means that the information JPDA filter reduces to classical information filter for STT without any clutters.

In MTT over a sensor network, each sensor node orders its estimated tracks differently and therefore track-to-track association is required to associate the tracks from different sensors that represent the i th target [125]. Typical track-to-track association utilises the so-called MDA formulation. If only two sensors are utilised in fusion, the MDA problem reduces to a classical 2D assignment problem, which can be efficiently solved by the well-known Hungarian algorithm. When the number of sensor nodes is larger than two, the track-to-track association MDA problem becomes NP-hard, which essentially coincides with the measurement-to-track association MDA in MHT². Therefore, any algorithms that suggested to solve the measurement-to-track MDA problem for MHT can be utilised in track-to-track association as well. There are a number of elegant choices for solving the MDA problem in combinatorial optimisation by reformulating the problem as a network flow [126] or using approximate Lagrangian relaxation approach [65, 127, 66]. Considering the balance between accuracy and efficiency, the Lagrangian relaxation method is utilised in this chapter.

After finding the same source origin of local tracks, information fusion can be performed based on the property of information filter: incorporating additional information from other sensors could be achieved by summation of the corresponding information terms. This implies that the optimal centralised JPDA over N sensors

²The only difference between track-to-track and measurement-to-track MDA problems is the assignment cost. The details will be discussed in the next chapter.

can be implemented as

$$\begin{aligned} \mathbf{Y}_{k|k}^{i,c} &= \mathbf{Y}_{k|k-1}^i + \sum_{l=1}^N \bar{\Omega}_{l,k}^i \\ \mathbf{x}_{k|k}^{i,c} &= \left(\mathbf{Y}_{k|k-1}^i + \sum_{l=1}^N \bar{\Omega}_{l,k}^i \right)^{-1} \left[\mathbf{y}_{k|k-1}^i + \sum_{l=1}^N (\delta_{l,k}^i + \beta_{l,0}^i \bar{\Omega}_{l,k}^i \mathbf{x}_{k|k-1}^i) \right] \end{aligned} \quad (6.11)$$

It is clear that that centralised estimation requires full information of all sensors. Considering the fact that each sensor usually can only communicate with its neighbours due to communication limit, this chapter will develop distributed implementation algorithms based on average consensus algorithm to recover the performance of the centralised estimation (6.11).

6.3 Distributed JPDA Filter Using Consensus on Measurement and Consensus on Information

This section extends conventional fusion strategies, e.g., consensus on measurement and consensus on information, to MTT scenarios on the basis of JPDA. Before providing the main results, the average consensus algorithm is briefly reviewed first.

6.3.1 Average Consensus

To perform estimation fusion in a distributed way, the concept of average consensus is adopted here. The average consensus algorithm is used to obtain the mean value of the information of all sensors in a distributed way without all-to-all communications; thus consensus-based distributed estimation can be applied to any generic connected sensor network. For the multi-sensor system, we use a connected undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to represent the communication topology, where $\mathcal{V} = \{\nu_1, \nu_2, \dots, \nu_N\}$ is a set of vertices that represent N sensors and $\mathcal{E} = \{(i, j) \in \mathcal{V} \times \mathcal{V}\}$ is a set of edges that stand for the relationship between two neighbouring sensors in the topology. If two sensors (i and j) are adjacent, namely, they can communicate with each other, then $(\nu_i, \nu_j) \in \mathcal{E}$. Denote $\mathcal{N}_i$ as the set of the neighbours (locally connected) of sensor i including sensor i . Denote a_l as the available information from the l th sensor and a_l is initialised as $a_l(0)$. Then, the distributed average consensus algorithm [18, 19]

at the m th iteration is defined as

$$a_l(m) = \sum_{l' \in \mathcal{N}_l} \pi_{l,l'} a_{l'}(m-1) \quad (6.12)$$

where $\pi_{l,l'}$ is the consensus gain, which satisfies the condition $\sum_{l' \in \mathcal{N}_l} \pi_{l,l'} = 1$ with $\pi_{l,l'} \geq 0$. The convergence rate of average consensus algorithm depends on the consensus gain and the algebraic connectivity of graph $\mathcal{G}$. Typical choices of $\pi_{l,l'}$ that guarantee the stability of the consensus phase are Metropolis weight and maximum-degree weight [128].

Lemma 6.1. [18, 19] *Under the assumption that the sensor network is connected, by running the iterative algorithm as in Eq. (6.12), the information of all sensors asymptotically converges to the initial average value as*

$$\lim_{m \rightarrow \infty} a_l(m) = \frac{1}{N} \sum_{l'=1}^N a_{l'}(0) \quad (6.13)$$

6.3.2 Distributed JPDA Filter Using Consensus on Measurement

Based on the principle of CM, DJPDA-CM aims to compute the summations $\sum_{l=1}^N \bar{\Omega}_{l,k}^i$, $\sum_{l=1}^N \Omega_{l,k}^i$ and $\sum_{l=1}^N (\delta_{l,k}^i + \beta_{l,0}^i \Omega_{l,k}^i \mathbf{x}_{l,k|k-1}^i)$ in Eq. (6.11) through a distributed way to recover the performance of centralised JPDA filter. Since these three terms are measurement-related, this fusion strategy is therefore called consensus on measurement. Define consensus variables $\mathbf{V}_{l,k}^i$, $\mathbf{G}_{l,k}^i$, $\mathbf{v}_{l,k}^i$, which are initialised as

$$\mathbf{V}_{l,k}^i(0) = \bar{\Omega}_{l,k}^i, \quad \mathbf{G}_{l,k}^i(0) = \Omega_{l,k}^i, \quad \mathbf{v}_{l,k}^i(0) = \delta_{l,k}^i + \beta_{l,0}^i \Omega_{l,k}^i \mathbf{x}_{l,k|k-1}^i \quad (6.14)$$

By sharing $\mathbf{V}_{l,k}^i$, $\mathbf{G}_{l,k}^i$, $\mathbf{v}_{l,k}^i$ with locally connected sensors through L iterations at every time instant, the measurement update of DJPDA-CM is given by

$$\begin{aligned} \mathbf{Y}_{l,k|k}^i &= \mathbf{Y}_{l,k|k-1}^i + N \mathbf{V}_{l,k}^i(L) \\ \mathbf{x}_{l,k|k}^i &= (\mathbf{Y}_{l,k|k-1}^i + N \mathbf{G}_{l,k}^i(L))^{-1} (\mathbf{y}_{l,k|k-1}^i + N \mathbf{v}_{l,k}^i(L)) \end{aligned} \quad (6.15)$$

where $\mathbf{V}_{l,k}^i(L)$, $\mathbf{G}_{l,k}^i(L)$, $\mathbf{v}_{l,k}^i(L)$ denote the values of $\mathbf{V}_{l,k}^i$, $\mathbf{G}_{l,k}^i$, $\mathbf{v}_{l,k}^i$ after performing L steps of average consensus.

The detailed implementation of the proposed DJPDA-CM algorithm is summarised in Algorithm 6.1. The following lemma analyses the asymptotic performance of

DJPDA-CM.

Lemma 6.2. *Under the assumption that the sensor network is connected with previously converged local estimates, e.g., $\mathbf{x}_{l,k-1|k-1}^i = \mathbf{x}_{k-1|k-1}^{i,c}$, $\mathbf{Y}_{l,k-1|k-1}^i = \mathbf{Y}_{k-1|k-1}^{i,c}$, DJPDA-CM (6.15) will asymptotically converge to the centralised JPDA (6.11) at current time instant.*

Proof. According to Lemma 6.1, applying average consensus algorithm to $\mathbf{V}_{l,k}^i$, $\mathbf{G}_{l,k}^i$, $\mathbf{v}_{l,k}^i$ gives

$$\begin{aligned}\lim_{m \rightarrow \infty} \mathbf{V}_{l,k}^i(m) &= \frac{1}{N} \sum_{l'=1}^N \bar{\Omega}_{l',k}^i \\ \lim_{m \rightarrow \infty} \mathbf{G}_{l,k}^i(m) &= \frac{1}{N} \sum_{l'=1}^N \Omega_{l',k}^i \\ \lim_{m \rightarrow \infty} \mathbf{v}_{l,k}^i(m) &= \frac{1}{N} \sum_{l'=1}^N (\delta_{l',k}^i + \beta_{l',0}^i \Omega_{l',k}^i \mathbf{x}_{l',k|k-1}^i)\end{aligned}\tag{6.16}$$

Therefore, the measurement update of DJPDA-CM with infinite number of iterations can be formulated as

$$\begin{aligned}\mathbf{Y}_{l,k|k}^i &= \mathbf{Y}_{l,k|k-1}^i + \sum_{l'=1}^N \bar{\Omega}_{l',k}^i \\ \mathbf{x}_{l,k|k}^i &= \left(\mathbf{Y}_{l,k|k-1}^i + \sum_{l'=1}^N \Omega_{l',k}^i \right)^{-1} \left[\mathbf{y}_{l,k|k-1}^i + \sum_{l'=1}^N (\delta_{l',k}^i + \beta_{l',0}^i \Omega_{l',k}^i \mathbf{x}_{l',k|k-1}^i) \right]\end{aligned}\tag{6.17}$$

Under the assumption that the previous local estimates are equal to the centralised fusion as $\mathbf{x}_{l,k-1|k-1}^i = \mathbf{x}_{k-1|k-1}^{i,c}$, $\mathbf{Y}_{l,k-1|k-1}^i = \mathbf{Y}_{k-1|k-1}^{i,c}$, it is straightforward to verify that $\mathbf{x}_{l,k|k}^i = \mathbf{x}_{k|k}^{i,c}$, $\mathbf{Y}_{l,k|k}^i = \mathbf{Y}_{k|k}^{i,c}$ from Eq (6.17). $\square$

Remark 6.1. *Lemma 6.2 reveals that the advantage of DJPDA-CM is that it is asymptotically optimal at each time instant provided that the priors are converged. However, since only finite number of consensus iterations is tractable in practice, convergence will not be fully achieved. The performance of DJPDA-CM with small number of consensus iterations will degrade significantly. To see this, define $\mathbf{\Pi}$ as the consensus matrix, whose elements are the consensus gains $\pi_{l,l'}$ and let $\pi_{l,l'}^m \in \mathbf{\Pi}^m$ with $\mathbf{\Pi}^m$ being the m th power of matrix $\mathbf{\Pi}$. Further, define $\mathcal{N}_l^m$ as the set of sensor nodes that the l th sensor can be connected within m hops. Then, only when $l' \in \mathcal{N}_l^m$, $\pi_{l,l'}^m \neq 0$. This means that only when the observability of sensor set*

Algorithm 6.1 DJPDA-CM for target i at sensor node l at time step k

Input: Previous estimation $\{\mathbf{x}_{l,k-1|k-1}^i, \mathbf{P}_{l,k-1|k-1}^i\}$, received measurements $\mathcal{Z}_k$

Output: Current estimation $\{\mathbf{x}_{l,k|k}^i, \mathbf{P}_{l,k|k}^i\}$

(1) Prediction:

$$\begin{aligned}\mathbf{x}_{l,k|k-1}^i &= \mathbf{F}_{l,k-1}^i \mathbf{x}_{l,k-1|k-1}^i \\ \mathbf{P}_{l,k|k-1}^i &= \mathbf{F}_{l,k-1}^i \mathbf{P}_{l,k-1|k-1}^i (\mathbf{F}_{l,k-1}^i)^T + \mathbf{Q}_{k-1}^i \\ \mathbf{Y}_{l,k|k-1}^i &= (\mathbf{P}_{l,k|k-1}^i)^{-1}, \mathbf{y}_{l,k|k-1}^i = (\mathbf{P}_{l,k|k-1}^i)^{-1} \mathbf{x}_{l,k|k-1}^i\end{aligned}$$

(2) Compute the measurement-related terms: $\bar{\boldsymbol{\Omega}}_{l,k}^i, \boldsymbol{\Omega}_{l,k}^i, \boldsymbol{\delta}_{l,k}^i$

(3) Compute the initial values of consensus variables:

$$\mathbf{V}_{l,k}^i(0) = \bar{\boldsymbol{\Omega}}_{l,k}^i, \quad \mathbf{G}_{l,k}^i(0) = \boldsymbol{\Omega}_{l,k}^i, \quad \mathbf{v}_{l,k}^i(0) = \boldsymbol{\delta}_{l,k}^i + \beta_{l,0}^i \boldsymbol{\Omega}_{l,k}^i \mathbf{x}_{l,k|k-1}^i$$

(4) **for** $m = 0, 1, \dots, L$ **do**

Broadcast information $\mathbf{V}_{l,k}^i, \mathbf{G}_{l,k}^i, \mathbf{v}_{l,k}^i$ to locally connected sensor nodes

Applying average consensus algorithm to $\mathbf{V}_{l,k}^i, \mathbf{G}_{l,k}^i, \mathbf{v}_{l,k}^i$

(5) Measurement update:

$$\begin{aligned}\mathbf{Y}_{l,k|k}^i &= \mathbf{Y}_{l,k|k-1}^i + N \mathbf{V}_{l,k}(L) \\ \mathbf{x}_{l,k|k}^i &= (\mathbf{Y}_{l,k|k-1}^i + N \mathbf{G}_{l,k}(L))^{-1} (\mathbf{y}_{l,k|k-1}^i + N \mathbf{v}_{l,k}(L)) \\ \mathbf{P}_{l,k|k}^i &= (\mathbf{Y}_{l,k|k}^i)^{-1}\end{aligned}$$

$\mathcal{N}_l^m$ is satisfied, the strategy of consensus on measurement is meaningful for data fusion since no local prior knowledge is utilised for fusion in DJPDA-CM. Therefore, DJPDA-CM constrains the posterior estimates as the prior estimates if the sensor and its neighbours cannot detect the target due to limited field-of-view.

Remark 6.2. As stated in Lemma 6.2, the convergence of DJPDA-CM to the centralised fusion requires the condition that the previous local estimates are converged. In real applications, however, only finite number of consensus iterations are acceptable, which means that $\mathbf{x}_{l,k-1|k-1}^i \neq \mathbf{x}_{k-1|k-1}^{i,c}$ and $\mathbf{Y}_{l,k-1|k-1}^i \neq \mathbf{Y}_{k-1|k-1}^{i,c}$. Therefore, all local estimates are auto-correlated during the fusion phase and thus DJPDA-CM suffers from the well-known auto-correlation problem.

6.3.3 Distributed JPDA Filter Using Consensus on Information

Apart from DJPDA-CM, the covariance intersection approach [121] suggests an alternative way to design a distributed JPDA filter, that is, consensus on information matrix and information vector. Let us rewrite the measurement update of information JPDA filter, given in Eq. (6.10), as

$$\begin{aligned} \mathbf{Y}_{l,k|k}^i &= \mathbf{Y}_{l,k|k-1}^i + \bar{\boldsymbol{\Omega}}_{l,k}^i \\ (\mathbf{Y}_{l,k|k-1}^i + \boldsymbol{\Omega}_{l,k}^i) \mathbf{x}_{l,k|k}^i &= \mathbf{y}_{l,k|k-1}^i + \boldsymbol{\delta}_{l,k}^i + \beta_{l,0}^i \boldsymbol{\Omega}_{l,k}^i \mathbf{x}_{l,k|k-1}^i \end{aligned} \quad (6.18)$$

Since covariance intersection utilises a convex combination of local estimates for data fusion, the consensus variables in DJPDA-CI are related to information matrix and information vector. More specifically, the consensus variables $\mathbf{W}_{l,k}^i$, $\mathbf{Q}_{l,k}^i$, $\mathbf{q}_{l,k}^i$ are initialised as

$$\begin{aligned} \mathbf{W}_{l,k}^i(0) &= \mathbf{Y}_{l,k|k-1}^i + \bar{\boldsymbol{\Omega}}_{l,k}^i, \quad \mathbf{Q}_{l,k}^i(0) = \mathbf{Y}_{l,k|k-1}^i + \boldsymbol{\Omega}_{l,k}^i \\ \mathbf{q}_{l,k}^i(0) &= \mathbf{y}_{l,k|k-1}^i + \boldsymbol{\delta}_{l,k}^i + \beta_{l,0}^i \boldsymbol{\Omega}_{l,k}^i \mathbf{x}_{l,k|k-1}^i \end{aligned} \quad (6.19)$$

After receiving $\mathbf{W}_{l,k}^i$, $\mathbf{Q}_{l,k}^i$, $\mathbf{q}_{l,k}^i$ from neighbours and performing L steps of average consensus at every time instant, the measurement update of DJPDA-CI at time instant k is given by

$$\begin{aligned} \mathbf{Y}_{l,k|k}^i &= \mathbf{W}_{l,k}(L) \\ \mathbf{x}_{l,k|k}^i &= [\mathbf{Q}_{l,k}(L)]^{-1} \mathbf{q}_{l,k}(L) \end{aligned} \quad (6.20)$$

The detailed implementation of the proposed DJPDA-CI algorithm is summarised in Algorithm 6.2. The following lemma analyses the asymptotical performance of DJPDA-CI.

Algorithm 6.2 DJPDA-CI for target i at sensor node l at time step k

Input: Previous estimation $\left\{ \mathbf{x}_{l,k-1|k-1}^i, \mathbf{P}_{l,k-1|k-1}^i \right\}$, received measurements $\mathcal{Z}_k$

Output: Current estimation $\left\{ \mathbf{x}_{l,k|k}^i, \mathbf{P}_{l,k|k}^i \right\}$

(1) Prediction:

$$\begin{aligned} \mathbf{x}_{l,k|k-1}^i &= \mathbf{F}_{l,k-1}^i \mathbf{x}_{l,k-1|k-1}^i \\ \mathbf{P}_{l,k|k-1}^i &= \mathbf{F}_{l,k-1}^i \mathbf{P}_{l,k-1|k-1}^i (\mathbf{F}_{l,k-1}^i)^T + \mathbf{Q}_{k-1}^i \\ \mathbf{Y}_{l,k|k-1}^i &= (\mathbf{P}_{l,k|k-1}^i)^{-1}, \mathbf{y}_{l,k|k-1}^i = (\mathbf{P}_{l,k|k-1}^i)^{-1} \mathbf{x}_{l,k|k-1}^i \end{aligned}$$

(2) Compute the measurement-related terms: $\bar{\boldsymbol{\Omega}}_{l,k}^i, \boldsymbol{\Omega}_{l,k}^i, \boldsymbol{\delta}_{l,k}^i$

(3) Compute the initial values of consensus variables:

$$\begin{aligned} \mathbf{W}_{l,k}^i(0) &= \mathbf{Y}_{l,k|k-1}^i + \bar{\boldsymbol{\Omega}}_{l,k}^i, \quad \mathbf{Q}_{l,k}^i(0) = \mathbf{Y}_{l,k|k-1}^i + \boldsymbol{\Omega}_{l,k}^i \\ \mathbf{q}_{l,k}^i(0) &= \mathbf{y}_{l,k|k-1}^i + \boldsymbol{\delta}_{l,k}^i + \beta_{l,0}^i \boldsymbol{\Omega}_{l,k}^i \mathbf{x}_{l,k|k-1}^i \end{aligned}$$

(4) **for** $m = 0, 1, \dots, L$ **do**

Broadcast information $\mathbf{W}_{l,k}^i, \mathbf{Q}_{l,k}^i, \mathbf{q}_{l,k}^i$ to locally connected sensor nodes

Applying average consensus algorithm to $\mathbf{W}_{l,k}^i, \mathbf{Q}_{l,k}^i, \mathbf{q}_{l,k}^i$

(5) Measurement update:

$$\begin{aligned} \mathbf{Y}_{l,k|k}^i &= \mathbf{W}_{l,k}^i(L) \\ \mathbf{x}_{l,k|k}^i &= [\mathbf{Q}_{l,k}^i(L)]^{-1} \mathbf{q}_{l,k}^i(L) \\ \mathbf{P}_{l,k|k}^i &= (\mathbf{Y}_{l,k|k}^i)^{-1} \end{aligned}$$

Lemma 6.3. *Under the assumption that the sensor network is connected, DJPDA-CI (6.20) cannot recover the performance of the centralised JPDA (6.11) even with infinite number of consensus iterations at current time instant.*

Proof. Using Lemma 6.1, the consensus variables $\mathbf{W}_{l,k}^i, \mathbf{Q}_{l,k}^i, \mathbf{q}_{l,k}^i$ asymptotically

converge to

$$\begin{aligned}
\lim_{m \rightarrow \infty} \mathbf{W}_{l,k}^i(m) &= \frac{1}{N} \sum_{l'=1}^N (\mathbf{Y}_{l',k|k-1}^i + \bar{\boldsymbol{\Omega}}_{l',k}^i) \\
\lim_{m \rightarrow \infty} \mathbf{Q}_{l,k}^i(m) &= \frac{1}{N} \sum_{l'=1}^N (\mathbf{Y}_{l',k|k-1}^i + \boldsymbol{\Omega}_{l',k}^i) \\
\lim_{m \rightarrow \infty} \mathbf{q}_{l,k}^i(m) &= \frac{1}{N} \sum_{l'=1}^N (\mathbf{y}_{l',k|k-1}^i + \boldsymbol{\delta}_{l',k}^i + \beta_{l',0}^i \boldsymbol{\Omega}_{l',k}^i \mathbf{x}_{l',k|k-1}^i)
\end{aligned} \tag{6.21}$$

Therefore, the measurement update of DJPDA-CI with infinite number of consensus iterations is determined as

$$\begin{aligned}
\mathbf{Y}_{l,k|k}^i &= \frac{1}{N} \sum_{l'=1}^N (\mathbf{Y}_{l',k|k-1}^i + \bar{\boldsymbol{\Omega}}_{l',k}^i) \\
\mathbf{x}_{l,k|k}^i &= \left[\sum_{l'=1}^N (\mathbf{Y}_{l',k|k-1}^i + \boldsymbol{\Omega}_{l',k}^i) \right]^{-1} \left[\sum_{l'=1}^N (\mathbf{y}_{l',k|k-1}^i + \boldsymbol{\delta}_{l',k}^i + \beta_{l',0}^i \boldsymbol{\Omega}_{l',k}^i \mathbf{x}_{l',k|k-1}^i) \right]
\end{aligned} \tag{6.22}$$

Equation (6.22) can be reformulated as

$$\begin{aligned}
\mathbf{Y}_{l,k|k}^i &= \frac{1}{N} \sum_{l'=1}^N \mathbf{Y}_{l',k|k-1}^i + \frac{1}{N} \sum_{l'=1}^N \bar{\boldsymbol{\Omega}}_{l',k}^i \\
\mathbf{x}_{l,k|k}^i &= \left(\frac{1}{N} \sum_{l'=1}^N \mathbf{Y}_{l',k|k-1}^i + \frac{1}{N} \sum_{l'=1}^N \boldsymbol{\Omega}_{l',k}^i \right)^{-1} \\
&\quad \times \left[\frac{1}{N} \sum_{l'=1}^N \mathbf{y}_{l',k|k-1}^i + \frac{1}{N} \sum_{l'=1}^N (\boldsymbol{\delta}_{l',k}^i + \beta_{l',0}^i \boldsymbol{\Omega}_{l',k}^i \mathbf{x}_{l',k|k-1}^i) \right]
\end{aligned} \tag{6.23}$$

which differs from Eq. (6.11). Therefore, DJPDA-CI is not asymptotically optimal. $\square$

Remark 6.3. From Eq. (6.18), we know that the term $\mathbf{y}_{l,k|k-1}^i + \boldsymbol{\delta}_{l,k}^i + \beta_{l,0}^i \boldsymbol{\Omega}_{l,k}^i \mathbf{x}_{l,k|k-1}^i$ corresponds to local estimate weighted by its updated information matrix. Then, it is clear that, in every consensus iteration of DJPDA-CI, the l th sensor leverages a convex combination of local estimates in $\mathcal{N}_l$ with suitable weights $\pi_{l,\nu}$. This is advantageous in ensuring bounded estimation errors for any number (even with single one) of consensus steps due to the consistency property of CI. Another benefit of DJPDA-CI is that it is robust against the auto-correlation among local estimates due to the property of CI [121]. Apart from its advantages, Lemma 6.3 demon-

strates that DJPDA-CI is a conservative fusion algorithm as the information terms $\sum_{l=1}^N \bar{\Omega}_{l,k}^i$, $\sum_{l=1}^N \Omega_{l,k}^i$ and $\sum_{l=1}^N \left(\delta_{l,k}^i + \beta_{l,0}^i \Omega_{l,k}^i \mathbf{x}_{k|k-1}^i \right)$ are underweighted by a scalar $1/N$, compared to the centralised solution (6.11).

Remark 6.4. Notice that DJPDA-CI with single consensus iteration, i.e., $L = 1$ reduces to the general CI applied to locally connected sensors. By utilising multiple consensus iteration steps, i.e., $L > 1$, the information of each sensor can be transmitted to more local sensors, thus improving the overall performance. Different from STT, the distributed CI of JPDA requires performing consensus on two different information matrices, e.g., $\mathbf{W}_{l,k}^i$ and $\mathbf{Q}_{l,k}^i$. This fact can be attributed to the intrinsic data association uncertainty property of MTT.

6.4 Distributed JPDA Filter Using Hybrid Fusion Strategy

6.4.1 Distributed JPDA Filter Using Hybrid Consensus

Comparing DJPDA-CM and DJPDA-CI, we can observe that these two algorithms have complementary properties: DJPDA-CM is asymptotically optimal but its performance degrades significantly when the number of consensus iterations is small; DJPDA-CI is beneficial for ensuring the consistency of fused estimates but cannot recover the performance of centralised JPDA. Furthermore, DJPDA-CM reveals that recovering the performance of centralised JPDA requires the knowledge of network size, e.g., the total number of sensor nodes. In practice, an unexpected sensor failure might happen and this changes the total number of nodes. Motivated by these observations, this chapter proposes a fully distributed JPDA filter using a hybrid consensus strategy to fully exploits the benefits of both DJPDA-CM and DJPDA-CI without the knowledge of N .

Let us define the consensus variables $\mathbf{U}_{l,k}^i$, $\mathbf{u}_{l,k}^i$, $\mathbf{V}_{l,k}^i$, $\mathbf{G}_{l,k}^i$, $\mathbf{v}_{l,k}^i$, $b_{l,k}$, $c_{l,k}$, which are initialised as

$$\begin{aligned} \mathbf{U}_{l,k}^i(0) &= \mathbf{Y}_{l,k|k-1}^i, \quad \mathbf{u}_{l,k}^i(0) = \mathbf{y}_{l,k|k-1}^i \\ \mathbf{V}_{l,k}^i(0) &= \bar{\Omega}_{l,k}^i, \quad \mathbf{G}_{l,k}^i(0) = \Omega_{l,k}^i, \quad \mathbf{v}_{l,k}^i(0) = \delta_{l,k}^i + \beta_{l,0}^i \Omega_{l,k}^i \mathbf{x}_{l,k|k-1}^i \\ b_{l,k}(0) &= 1, \quad c_{l',k}(0) = 1 \ (\exists l'), \quad c_{l,k}(0) = 0 \ (l \neq l') \end{aligned} \quad (6.24)$$

where $b_{l,k}$, $c_{l,k}$ are utilised to estimate the network size and $\exists l'$ denotes that there

exists only one sensor node l' .

The proposed hybrid fusion scheme consists of three different types of data fusion: (1) Consensus on prior $\mathbf{U}_{l,k}^i, \mathbf{u}_{l,k}^i$; (2) Consensus on measurement $\mathbf{V}_{l,k}^i, \mathbf{G}_{l,k}^i, \mathbf{v}_{l,k}^i$; and (3) Consensus on counting variables $b_{l,k}, c_{l,k}$. The measurement update of DJPDA-HC at time instant k with finite L consensus iterations is given by

$$\begin{aligned} \mathbf{Y}_{l,k|k}^i &= \mathbf{U}_{l,k}(L) + \frac{b_{l,k}(L)}{c_{l,k}(L)} \mathbf{V}_{l,k}(L) \\ \mathbf{x}_{l,k|k}^i &= \left[\mathbf{U}_{l,k}(L) + \frac{b_{l,k}(L)}{c_{l,k}(L)} \mathbf{G}_{l,k}(L) \right]^{-1} \left(\mathbf{u}_{l,k}(L) + \frac{b_{l,k}(L)}{c_{l,k}(L)} \mathbf{v}_{l,k}(L) \right) \end{aligned} \quad (6.25)$$

Note from Eq. (6.24) that implementing DJPDA-HC requires selecting a special sensor node l that is initialised as $b_{l,k}(0) = 1, c_{l,k}(0) = 1$. Obviously, different choices of this special sensor affect the overall estimation performance. To address this problem, the random max consensus algorithm [129] is utilised here as an alternative way to find the index of the special node in a probabilistic manner. Specifically, define consensus variables $b_{l,k}, c_{l,k}, \lambda_{l,k}$ which are initialised as $b_{l,k}(0) = 1, c_{l,k}(0) = 1, \lambda_{l,k}(0) \sim \mathcal{U}(0, 1)$, where $\mathcal{U}(0, 1)$ denotes the uniform distribution with lower and upper bounds being, respectively, 0 and 1. This random initialisation guarantees the diversity of the proposed algorithm: every sensor node has the possibility to become the special one. At each average consensus iteration, $\lambda_{l,k}$ is updated as the output of random max consensus algorithm, denoted as $\lambda_{l,\max}$, which is the maximum value of $\lambda_{l',k}, \forall l' \in \mathcal{N}_l$. Then, the first time that $\lambda_{l,k}(m+1) \neq \lambda_{l,k}(m)$, we subtract 1 from $c_{l,k}(m+1)$. Since the random max consensus algorithm will eventually find the sensor node that has the largest value of $\lambda_{l,k}$ with certain m' steps [129], we have $\sum_{l=1}^N c_{l,k}(m) = 1$ when $m > m'$. Therefore, the total number of sensors can be approximated by $N = b_{l,k}(L)/c_{l,k}(L)$, with $L > m'$ consensus steps.

The detailed implementation of the proposed DJPDA-CI algorithm is summarised in Algorithm 6.3. The following lemma analyses the asymptotical performance of DJPDA-HC.

Lemma 6.4. *Under the assumption that the sensor network is connected, DJPDA-HC (6.25) will asymptotically converge to the centralised JPDA (6.11) at current time instant.*

Algorithm 6.3 DJPDA-HC for target i at sensor node l at time step k

Input: Previous estimation $\{\mathbf{x}_{l,k-1|k-1}^i, \mathbf{P}_{l,k-1|k-1}^i\}$, received measurements $\mathcal{Z}_k$

Output: Current estimation $\{\mathbf{x}_{l,k|k}^i, \mathbf{P}_{l,k|k}^i\}$

(1) Prediction:

$$\begin{aligned}\mathbf{x}_{l,k|k-1}^i &= \mathbf{F}_{l,k-1}^i \mathbf{x}_{l,k-1|k-1}^i \\ \mathbf{P}_{l,k|k-1}^i &= \mathbf{F}_{l,k-1}^i \mathbf{P}_{l,k-1|k-1}^i (\mathbf{F}_{l,k-1}^i)^T + \mathbf{Q}_{k-1}^i \\ \mathbf{Y}_{l,k|k-1}^i &= (\mathbf{P}_{l,k|k-1}^i)^{-1}, \mathbf{y}_{l,k|k-1}^i = (\mathbf{P}_{l,k|k-1}^i)^{-1} \mathbf{x}_{l,k|k-1}^i\end{aligned}$$

(2) Compute the measurement-related terms: $\bar{\boldsymbol{\Omega}}_{l,k}^i, \boldsymbol{\Omega}_{l,k}^i, \boldsymbol{\delta}_{l,k}^i$

(3) Compute the initial values of consensus variables:

$$\begin{aligned}\mathbf{U}_{l,k}^i(0) &= \mathbf{Y}_{l,k|k-1}^i, \quad \mathbf{u}_{l,k}^i(0) = \mathbf{y}_{l,k|k-1}^i \\ \mathbf{V}_{l,k}^i(0) &= \bar{\boldsymbol{\Omega}}_{l,k}^i, \quad \mathbf{G}_{l,k}^i(0) = \boldsymbol{\Omega}_{l,k}^i, \quad \mathbf{v}_{l,k}^i(0) = \boldsymbol{\delta}_{l,k}^i + \beta_{l,0}^i \boldsymbol{\Omega}_{l,k}^i \mathbf{x}_{l,k|k-1}^i \\ b_{l,k}(0) &= 1, \quad c_{l,k}(0) = 1, \quad \lambda_{l,k}(0) \sim \mathcal{U}(0, 1)\end{aligned}$$

(4) Initialise the subtraction flag for sensor node counting algorithm: $\alpha_{l,k} = 1$

(5) **for** $m = 0, 1, \dots, L$ **do**

Broadcast information $\mathbf{U}_{l,k}^i, \mathbf{u}_{l,k}^i, \mathbf{V}_{l,k}^i, \mathbf{G}_{l,k}^i, \mathbf{v}_{l,k}^i, b_{l,k}, c_{l,k}, \lambda_{l,k}$ to locally connected sensor nodes

Apply average consensus algorithm to $\mathbf{U}_{l,k}^i, \mathbf{u}_{l,k}^i, \mathbf{V}_{l,k}^i, \mathbf{G}_{l,k}^i, \mathbf{v}_{l,k}^i, b_{l,k}, c_{l,k}$

Apply random max consensus algorithm to $\lambda_{l,k}$ and update $\lambda_{l,k}(m+1)$ as

$\lambda_{l,\max}$

if $\lambda_{l,k}(m+1) \neq \lambda_{l,k}(m)$ and $\alpha_{l,k} = 1$ **do**

$$c_{l,k}(m+1) = c_{l,k}(m) - 1$$

$$\alpha_{l,k} = 0$$

(6) Measurement update:

$$\begin{aligned}\mathbf{Y}_{l,k|k}^i &= \mathbf{U}_{l,k}(L) + \frac{b_{l,k}(L)}{c_{l,k}(L)} \mathbf{V}_{l,k}(L) \\ \mathbf{x}_{l,k|k}^i &= \left[\mathbf{U}_{l,k}(L) + \frac{b_{l,k}(L)}{c_{l,k}(L)} \mathbf{G}_{l,k}(L) \right]^{-1} \left(\mathbf{u}_{l,k}(L) + \frac{b_{l,k}(L)}{c_{l,k}(L)} \mathbf{v}_{l,k}(L) \right) \\ \mathbf{P}_{l,k|k}^i &= (\mathbf{Y}_{l,k|k}^i)^{-1}\end{aligned}$$

Proof. Applying average consensus algorithm to $\mathbf{U}_{l,k}^i$, $\mathbf{u}_{l,k}^i$, $\mathbf{V}_{l,k}^i$, $\mathbf{G}_{l,k}^i$, $\mathbf{v}_{l,k}^i$, we have

$$\begin{aligned}
\lim_{m \rightarrow \infty} \mathbf{U}_{l,k}^i(m) &= \frac{1}{N} \sum_{l'=1}^N \mathbf{Y}_{l',k|k-1}^i \\
\lim_{m \rightarrow \infty} \mathbf{v}_{l,k}^i(m) &= \frac{1}{N} \sum_{l'=1}^N \mathbf{y}_{l',k|k-1}^i \\
\lim_{m \rightarrow \infty} \mathbf{V}_{l,k}^i(m) &= \frac{1}{N} \sum_{l'=1}^N \bar{\Omega}_{l',k}^i \\
\lim_{m \rightarrow \infty} \mathbf{G}_{l,k}^i(m) &= \frac{1}{N} \sum_{l'=1}^N \Omega_{l',k}^i \\
\lim_{m \rightarrow \infty} \mathbf{v}_{l,k}^i(m) &= \frac{1}{N} \sum_{l'=1}^N (\delta_{l',k}^i + \beta_{l',0}^i \Omega_{l',k}^i \mathbf{x}_{l',k|k-1}^i)
\end{aligned} \tag{6.26}$$

Since the random max consensus algorithm will eventually find the special sensor node with largest value of $\lambda_{l,k}$, average consensus algorithm guarantees

$$\begin{aligned}
\lim_{m \rightarrow \infty} b_{l,k}(m) &= \frac{1}{N} \sum_{l'=1}^N b_{l',k}(0) = 1 \\
\lim_{m \rightarrow \infty} c_{l,k}(m) &= \frac{1}{N} \sum_{l'=1}^N c_{l',k}(0) = \frac{1}{N}
\end{aligned} \tag{6.27}$$

Substituting Eqs. (6.26) and (6.27) into Eq. (6.25) gives the proposed DJPDA-HC with infinite consensus iterations as

$$\begin{aligned}
\mathbf{Y}_{l,k|k}^i &= \frac{1}{N} \sum_{l'=1}^N \mathbf{Y}_{l',k|k-1}^i + \sum_{l'=1}^N \bar{\Omega}_{l',k}^i \\
\mathbf{x}_{l,k|k}^i &= \left(\frac{1}{N} \sum_{l'=1}^N \mathbf{Y}_{l',k|k-1}^i + \sum_{l'=1}^N \Omega_{l',k}^i \right)^{-1} \\
&\quad \times \left[\frac{1}{N} \sum_{l'=1}^N \mathbf{y}_{l',k|k-1}^i + \sum_{l'=1}^N (\delta_{l',k}^i + \beta_{l',0}^i \Omega_{l',k}^i \mathbf{x}_{l',k|k-1}^i) \right]
\end{aligned} \tag{6.28}$$

Since consensus on priors guarantees that all local sensors have the same priori estimates, e.g., $\mathbf{Y}_{l,k|k-1}^i = \frac{1}{N} \sum_{l'=1}^N \mathbf{Y}_{l',k|k-1}^i$ and $\mathbf{y}_{l,k|k-1}^i = \frac{1}{N} \sum_{l'=1}^N \mathbf{y}_{l',k|k-1}^i$, it is immediate to see that DJPDA-HC with infinite number of consensus iterations can recover the performance of centralised JPDA. That is, DJPDA-HC is asymptotically optimal. $\square$

Remark 6.5. *It is worth to point out that, with small number of iterations, $b_{l,k}(L)$*

is very close to $c_{l,k}(L)$ due to the same initialisations. For example, setting $L = 1$ for a sparse sensor network results in $b_{l,k}(L) \approx c_{l,k}(L)$. Under this condition, DJPDA-HC shows similar performance as DJPDA-CI. With the increase of L , we have $b_{l,k}(L) / c_{l,k}(L) \approx N$ and DJPDA-HC gradually converges to optimal centralised JPDA, similar to DJPDA-CM. Therefore, DJPDA-HC fully exploits the benefits of both DJPDA-CM, e.g., global convergence to the optimal centralised solution with infinite consensus iterations, and DJPDA-CI, e.g., preserving local consistency when the consensus horizon is limited.

6.4.2 Algorithm Analysis

This subsection analyses the differences between DJPDA-CM, DJPDA-CI and DJPDA-HC theoretically. For simplicity and convenience of analysis, the measurement updates of DJPDA-CM, DJPDA-CI and DJPDA-HC with a single consensus iteration, e.g., $L = 1$, are derived from Eqs. (6.15), (6.20) and (6.25) as

- DJPDA-CM:

$$\begin{aligned} \mathbf{Y}_{l,k|k}^i &= \mathbf{Y}_{l,k|k-1}^i + N \sum_{l' \in \mathcal{N}_l} \pi_{l,l'} \bar{\boldsymbol{\Omega}}_{l',k}^i \\ \mathbf{x}_{l,k|k}^i &= \left(\mathbf{Y}_{l,k|k-1}^i + N \sum_{l' \in \mathcal{N}_l} \pi_{l,l'} \boldsymbol{\Omega}_{l',k}^i \right)^{-1} \\ &\quad \times \left[\mathbf{y}_{l,k|k-1}^i + N \sum_{l' \in \mathcal{N}_l} \pi_{l,l'} \left(\boldsymbol{\delta}_{l',k}^i + \beta_{l',0}^i \boldsymbol{\Omega}_{l',k}^i \mathbf{x}_{l',k|k-1}^i \right) \right] \end{aligned} \quad (6.29)$$

- DJPDA-CI:

$$\begin{aligned} \mathbf{Y}_{l,k|k}^i &= \sum_{l' \in \mathcal{N}_l} \pi_{l,l'} \left(\mathbf{Y}_{l',k|k-1}^i + \bar{\boldsymbol{\Omega}}_{l',k}^i \right) \\ \mathbf{x}_{l,k|k}^i &= \left[\sum_{l' \in \mathcal{N}_l} \pi_{l,l'} \left(\mathbf{Y}_{l',k|k-1}^i + \boldsymbol{\Omega}_{l',k}^i \right) \right]^{-1} \\ &\quad \times \left[\sum_{l' \in \mathcal{N}_l} \pi_{l,l'} \left(\mathbf{y}_{l',k|k-1}^i + \boldsymbol{\delta}_{l',k}^i + \beta_{l',0}^i \boldsymbol{\Omega}_{l',k}^i \mathbf{x}_{l',k|k-1}^i \right) \right] \end{aligned} \quad (6.30)$$

- DJPDA-HC:

$$\begin{aligned}
\mathbf{Y}_{l,k|k}^i &= \sum_{l' \in \mathcal{N}_l} \pi_{l,l'} \mathbf{Y}_{l',k|k-1}^i + \frac{\sum_{l' \in \mathcal{N}_l} \pi_{l,l'} b_{l',k}}{\sum_{l' \in \mathcal{N}_l} \pi_{l,l'} c_{l',k}} \sum_{l' \in \mathcal{N}_l} \pi_{l,l'} \bar{\boldsymbol{\Omega}}_{l',k}^i \\
\mathbf{x}_{l,k|k}^i &= \left(\sum_{l' \in \mathcal{N}_l} \pi_{l,l'} \mathbf{Y}_{l',k|k-1}^i + \frac{\sum_{l' \in \mathcal{N}_l} \pi_{l,l'} b_{l',k}}{\sum_{l' \in \mathcal{N}_l} \pi_{l,l'} c_{l',k}} \sum_{l' \in \mathcal{N}_l} \pi_{l,l'} \boldsymbol{\Omega}_{l',k}^i \right)^{-1} \\
&\quad \times \left[\sum_{l' \in \mathcal{N}_l} \pi_{l,l'} \mathbf{y}_{l',k|k-1}^i + \frac{\sum_{l' \in \mathcal{N}_l} \pi_{l,l'} b_{l',k}}{\sum_{l' \in \mathcal{N}_l} \pi_{l,l'} c_{l',k}} \sum_{l' \in \mathcal{N}_l} \pi_{l,l'} (\boldsymbol{\delta}_{l',k}^i + \beta_{l',0}^i \boldsymbol{\Omega}_{l',k}^i \mathbf{x}_{l',k|k-1}^i) \right]
\end{aligned} \tag{6.31}$$

From the measurement updates of DJPDA-CM, DJPDA-CI and DJPDA-HC, we have the following important observations:

- Comparing with DJPDA-CM (6.29), the prior information vectors between locally connected sensor nodes of DJPDA-CI and DJPDA-HC are weighted by their corresponding prior information matrix as $\sum_{l' \in \mathcal{N}_l} \pi_{l,l'} \mathbf{Y}_{l',k|k-1}^i \mathbf{x}_{l',k|k-1}^i$. This handles the issue of naive sensors, e.g., target is outside of the sensor's field-of-view, by placing less weight when receiving the information from a naive neighbour sensor. Therefore, the proposed DJPDA-HC is helpful in ensuring the consistency of local estimates. It is clear that the implementation of DJPDA-CM requires the global information, e.g., the total number of sensors, whereas DJPDA-HC dynamically estimate N in a distributed way.
- Comparing with DJPDA-CI (6.30), the proposed DJPDA-HC remedies the drawback of measurement underestimation by multiplying a scaling factor $\sum_{l' \in \mathcal{N}_l} \pi_{l,l'} b_{l',k} / \sum_{l' \in \mathcal{N}_l} \pi_{l,l'} c_{l',k}$, providing the possibility of performance recovery of optimal centralised estimation. Note that the exact total information is the summation of measurements from all sensors, which is critical in recovering the performance of optimal centralised estimation.
- As the proposed DJPDA-HC algorithm requires more consensus variables for measurement update, the computational complexity of DJPDA-HC is higher than that of DJPDA-CM and DJPDA-CI. However, consensus on prior and consensus on measurement can be implemented in parallel to reduce the computational burden because prior estimates are independent of current measurements.

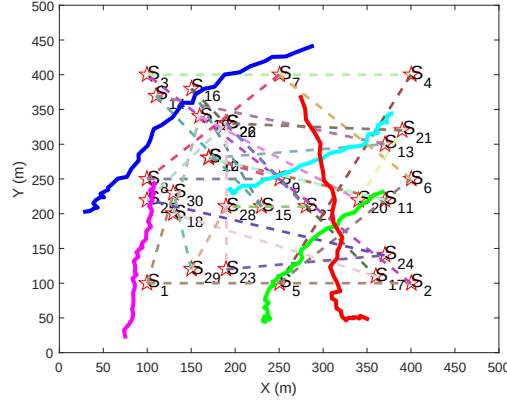


Figure 6.1: Snapshots of the considered scenario with pentagrams as sensors, colour dashed lines as network connection, and solid colour lines as target trajectories.

- For DJPDA-CM, DJPDA-CI and DJPDA-HC, it is worth to point out that both computational complexity and communication burden increase linearly with the increase in the number of consensus iteration steps L . For this reason, the parameter L should be selected as a suitable trade-off between cost and estimation performance in practical applications.

6.5 Numerical Simulations

This section presents a performance evaluation of the proposed DJPDA-CM, DJPDA-CM and DJPDA-HC algorithms using Monte-Carlo simulations and experiments.

6.5.1 Simulation Setup

The considered scenario considers 5 targets randomly moving in a $500m \times 500m$ rectangular area. The surveillance area is monitored by a sensor network with $N = 30$ sensors and each sensor's field-of-view is a $200m \times 200m$ rectangle. Each sensor in the network has the same degree of 2, which means that each sensor is locally connected with other two sensors. Under this condition, the communication topology of the sensor network is randomly generated. All sensors are randomly placed inside the $500m \times 500m$ rectangle to cover the entire surveillance area. One sample of the considered scenario is given in Fig. 6.1.

Each target's state is represented by a 4D vector, with 2D position and 2D velocity components. In estimation update, the system equation is assumed to be the well-

known CV model, where the system matrix $\mathbf{F}_k^i$ is given by

$$\mathbf{F}_k^i = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.32)$$

with $T_s = 1s$ being the sampling time. The variance of process noise of the considered constant velocity model is determined as

$$\mathbf{Q}_k^i = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.33)$$

For each sensor, if the target is located inside its field-of-view, the target-generated measurements are generated with a detection probability $P_D = 0.9$. Each sensor collects range as well as bearing measurements at regular time instants $t_k = kT_s$, $k \in \{1, 2, \dots, 40\}$, as

$$\mathbf{z}_k^i = \begin{bmatrix} \sqrt{(p_{x,k}^i - x_s^l)^2 + (p_{y,k}^i - y_s^l)^2} \\ \arctan\left(\frac{p_{y,k}^i - y_s^l}{p_{x,k}^i - x_s^l}\right) \end{bmatrix} + \mathbf{v}_k^i \quad (6.34)$$

where $(p_{x,k}^i, p_{y,k}^i)$ denotes the 2D inertial position of the i th target and (x_s^l, y_s^l) denotes the position of the l th sensor. The measurement noise is subject to a Gaussian white noise as $\mathbf{v}_k \sim \mathcal{N}(\cdot; 0, \mathbf{R}_k^i)$ with $\mathbf{R}_k^i = \text{diag}(\sigma_r^2, \sigma_a^2)$, $\sigma_r = 3m$, $\sigma_a = 0.5^\circ$. The clutter is assumed to be uniformly distributed in the surveillance region with its number being Poisson with 2 average returns per sensor at each scan. To accommodate the nonlinearity of measurements, the extended Kalman filter is applied in simulations.

For initialisation, the covariance matrix of the i th target at sensor node l is chosen as $\mathbf{P}_{l,0|-1}^i = \text{diag}(100, 100, 10, 10)$. The initial state estimates are generated from a Gaussian distribution around the true target state with the covariance $\mathbf{P}_{l,0|-1}^i$. Note that the starting point of each target is randomly generated inside the surveillance region at every Monte-Carlo run.

The number of consensus iterations is set as $L = 10$ and the Metropolis weights

[128] are leveraged for running average consensus algorithm as

$$\pi_{i,j} = \begin{cases} \frac{1}{\max\{|\mathcal{N}_i|, |\mathcal{N}_j|\}}, & \text{if } (i, j) \in \mathcal{E} \\ 1 - \sum_{(i,j) \in \mathcal{E}} \pi_{i,j}, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \quad (6.35)$$

where $|\mathcal{N}_i|$ denotes the cardinality, e.g., number of elements, of set $\mathcal{N}_i$.

In order to evaluate the performance of the proposed algorithms with different conditions, one parameter is varied while others are set as their aforementioned default values. The simulations are obtained over 1000 Monte-Carlo runs for each parameter setting.

6.5.2 Performance Metric

Let $\mathbf{x}_{l,k|k}^{i,j}$ denote the estimated state of the i th target at sensor node l at time instant k at the j th Monte Carlo run and $\mathbf{x}_k^{i,j}$ be the true state of the i th target at time instant k at the j th Monte Carlo run. The ME of position estimation and RMSE of position estimation at time instant k , averaged over M Monte-Carlo runs, N sensors and N_k targets, are defined as

$$\begin{aligned} \text{ME}_k^{\text{pos}} &= \frac{1}{MNN_k} \sum_{i=1}^{N_k} \sum_{l=1}^N \sum_{j=1}^M \left\| \mathbf{p}_{l,k|k}^{i,j} - \mathbf{p}_k^{i,j} \right\| \\ \text{RMSE}_k^{\text{pos}} &= \left(\frac{1}{MNN_k} \sum_{i=1}^{N_k} \sum_{l=1}^N \sum_{j=1}^M \left\| \mathbf{p}_{l,k|k}^{i,j} - \mathbf{p}_k^{i,j} \right\|^2 \right)^{\frac{1}{2}} \end{aligned} \quad (6.36)$$

where $\mathbf{p}_k^{i,j} = \mathbf{x}_k^{i,j} (1:2)$ and $\mathbf{p}_{l,k|k}^{i,j} = \mathbf{x}_{l,k|k}^{i,j} (1:2)$ are true and estimated target positions.

For performance evaluation of the proposed algorithms, the time averaged ME and RMSE are utilised. These two metrics are computed as

$$\begin{aligned} \text{ME}_{\text{avg}}^{\text{pos}} &= \frac{1}{T} \sum_{k=1}^T \text{ME}_k^{\text{pos}} \\ \text{RMSE}_{\text{avg}}^{\text{pos}} &= \frac{1}{T} \sum_{k=1}^T \text{RMSE}_k^{\text{pos}} \end{aligned} \quad (6.37)$$

where $T = 40$ is the total number of time instants during the tracking period.

6.5.3 Simulation Results

In consensus-based distributed estimation, information transmission via multiple communications among locally connected sensors are required and the performance is highly-related to the number of iteration steps L . In order to investigate the effect of the parameter L on filtering performance, Monte-Carlo simulations are performed with respect to different iterations $L = 1, 2, \dots, 20$. The simulation results of average target position estimation ME and RMSE are depicted in Fig. 6.2. From this figure, it can be noted that the performance of DJPDA-CM, DJPDA-CI and DJPDA-HC improves with the increase of iteration step L . The average MEs and RMSEs of DJPDA-CM, DJPDA-CI and DJPDA-HC all converge to some certain constants when the consensus iteration step becomes large enough. Note that DJPDA-CI generates more accurate estimation, compared to DJPDA-CM, with small number of consensus steps. This fact can be attributed to that DJPDA-CI utilises a convex combination of the prior estimates while DJPDA-CM only leverages the novel information for fusion. Fig. 6.2 also reveals that both DJPDA-CM and DJPDA-HC outperform DJPDA-CI with enough number of consensus iterations and guarantee a global convergence to recover the performance of the optimal centralised JPDA filter. This confirms the theoretical analysis shown in the previous section. With finite number of consensus iterations, the proposed DJPDA-HC provides better estimation performance, compared to both DJPDA-CM and DJPDA-CI, in terms of average ME and RMSE, demonstrating the advantages of DJPDA-HC algorithm. Notice that improvement in estimation can be obtained by increasing the number of consensus steps L . However, there is not much performance difference for the proposed DJPDA-HC with enough consensus steps, e.g., $L \geq 10$ in the considered setup. Typically, the communication rate is much faster than the sampling rate, meaning that certain consensus steps between two consecutive time instants can be ensured to guarantee the fusion performance.

Now, let us investigate the effect of the total number of sensors on tracking performance. Fig. 6.3 presents the Monte-Carlo simulation results of average target position estimation ME and RMSE with different number of sensors $N = 22, 24, \dots, 40$. Note that all sensors are placed randomly inside the surveillance area. Therefore, if one sensor is close to another, their range and angle observations share similar qualities. Intuitively, the total amount of information increases with more sensors,

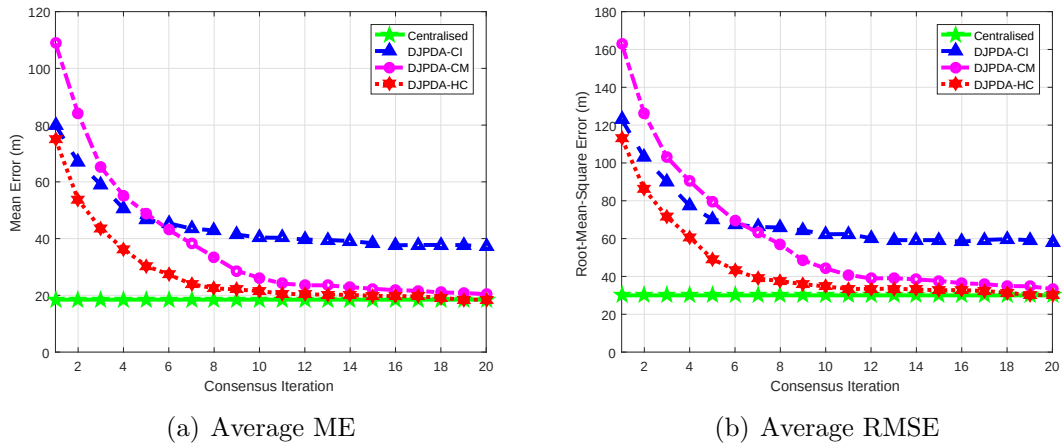


Figure 6.2: Average target position estimation ME and RMSE versus number of consensus iterations.

which should generate improved performance. However, more sensors with the same graph degree inevitably requires larger number of consensus iterations for convergence. Due to these contradictory facts, the performance of proposed algorithms do not show much difference for small number of sensors, e.g., $N \leq 30$. Compared to DJPDA-CI and DJPDA-HC, DJPDA-CM is more sensitive to the variation of total sensor number because this algorithm only utilises the available measurement information. It should be pointed out that the centralised JPDA filter also shows performance degradation with the increase of network size. This fact can be attributed to the increasing of the possibility of track-to-track association failure.

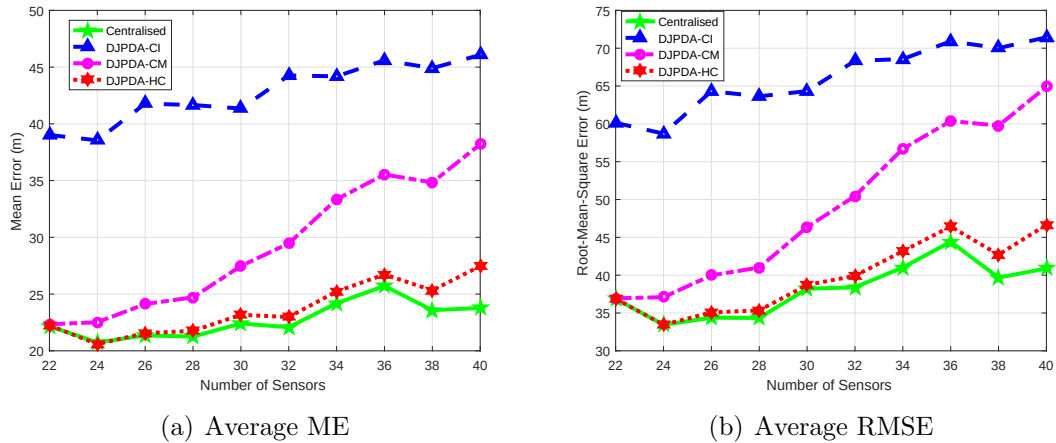


Figure 6.3: Average target position estimation ME and RMSE versus number of sensors.

In network-based sensing and tracking, it is clear that the sensor's available field-of-view would affect the overall estimation performance: if the sensor's field-of-view

is too narrow, then it cannot detect the target of interest. Fig. 6.4 compares the performance of different tracking algorithms with respect to different sensing range $50m, 100m, \dots, 400m$. Here, the sensor's field-of-view is defined as a rectangle, depending on the sensing range. For example, if the sensing range is $200m$, the sensor's field-of-view is a $200m \times 200m$ rectangle. From Fig. 6.4, it can be observed that, with narrow field-of-view, e.g., sensing range $\leq 100m$, both DJPDA-CI and DJPDA-HC outperform DJPDA-CM, demonstrating that fusing the prior information is helpful to preserve the consistency of local estimates. Note that the centralised JPDA filter also shows apparent performance degradation with narrow sensor's field-of-view. The reason is that most sensor nodes miss detect the targets due to narrow field-of-view. Therefore, only small amount of information can be utilised in the fusion centre for data integration. With longer sensing range, the performance of DJPDA-CM and DJPDA-HC improves significantly and converges to the optimal centralised JPDA since more information is available for data integration. With enough sensing range, e.g., $\geq 250m$, the performance of all tested distributed JPDA filters converges to certain steady-state performance. However, it is clear that DJPDA-CI is conservative and cannot converge to optimal centralised fusion.

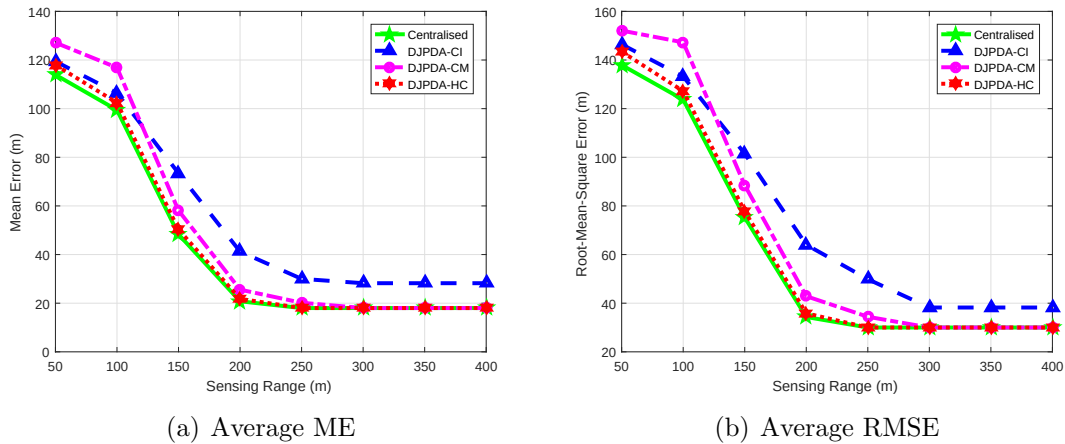


Figure 6.4: Average target position estimation ME and RMSE versus sensing range.

In MTT, the number of targets determines the complexity or size of the problem and greatly affects the tracking performance since data association becomes more challenging with the increase of problem complexity. Fig. 6.5 presents the Monte-Carlo simulation results of average target position estimation ME and RMSE with different number of targets $N_k = 3, 4, \dots, 8$. Unsurprisingly, the performance of all tested tracking algorithms degrades with more targets. The reason is that the chance of data association (measurement-to-track as well as track-to-track) failure

risers with the increase in the number of targets, which has an adverse effect in JPDA. Interestingly, the proposed DJPDA-HC exhibits very close performance as centralised JPDA and outperforms both DJPDA-CM and DJPDA-CI even with large number of targets.

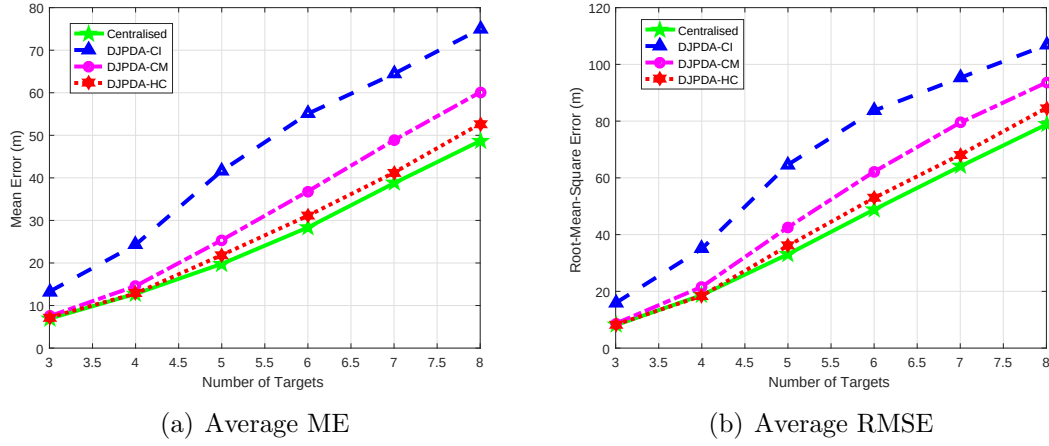


Figure 6.5: Average target position estimation ME and RMSE versus number of targets.

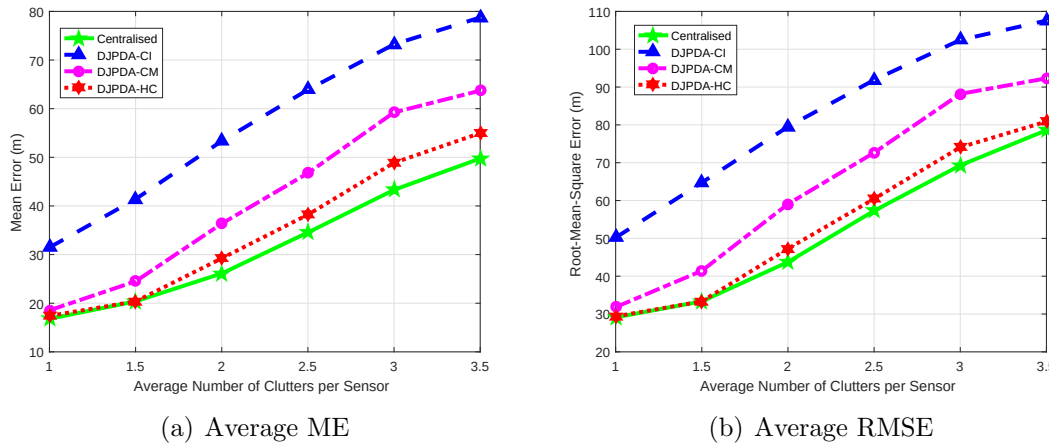


Figure 6.6: Average target position estimation ME and RMSE versus number of clutters per sensor.

High clutter rate is a typical characteristics of low-cost sensors, which would have adverse effect on the tracking performance. To investigate the robustness of different algorithms against the variation of clutter rate, Fig. 6.6 presents the Monte-Carlo simulation results of average target position estimation ME and RMSE with different number of clutters per sensor per frame $N_{FA} = 1, 1.5, \dots, 3.5$. Intuitively, the increase of clutter rate will give rise to high possibility of data association failure (measurement-to-track as well as track-to-track), thus the degradation in tracking

performance. This can be clearly observed from Fig. 6.6 for all tested algorithms. This figure also reveals that DJPDA-CM is more sensitive to other algorithms. Furthermore, the proposed DJPDA-HC algorithm is strongly robust even to a high clutter rate and shows very close performance as the centralised JPDA filter.

6.5.4 Experiments on Real-Life Data

To further validate the effectiveness of the proposed method, we evaluate it with publicly available multiple target tracking datasets. We choose the popular EPFL laboratory video sequences [130] for pedestrian tracking. The EPFL laboratory dataset, filmed from four different cameras, offers four pedestrians walking around in a room. Figure 6.7 presents the snapshots from these four different cameras.

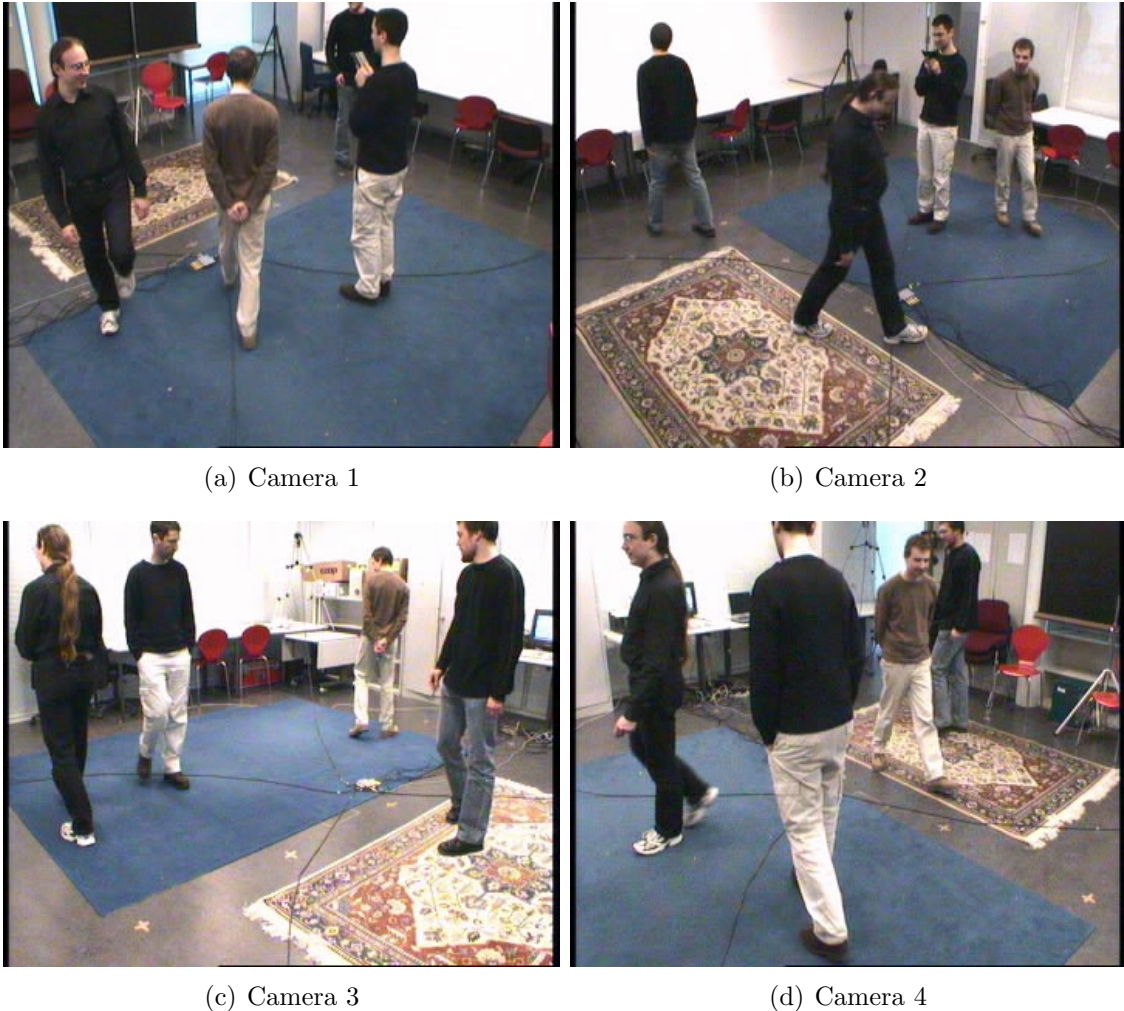


Figure 6.7: Snapshots of EPFL laboratory dataset from four different cameras.

In all experiments, the pedestrian position measurements are obtained by running

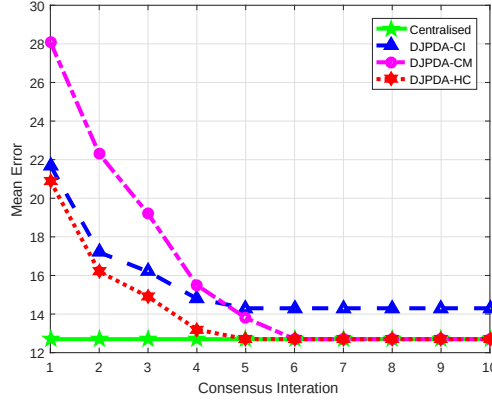


Figure 6.8: Mean position error averaged over four cameras and four persons.

the state-of-the-art discriminatively trained part-based model (DTPM) detector [131]. The detections are obtained on a post-processing basis for all collected data to allow for fair comparisons with exactly the same inputs, i.e., the DTPM detector is applied to all videos in an offline fashion to get the measurements of all frames. Each detection is represented by a bounding box with its centroid, length and width as the measurement information of each detected target. For valid fusion, all DJPDA algorithms are performed using a ground-based inertial coordinate. Under this condition, the nonlinear measurement model $h_{l,k}^i(\mathbf{x}_k^i)$ of the l th camera is a function of the camera calibration parameters [130]. The tracking performance are evaluated by comparing with the annotated ground truth, provided by [132]. The mean position error, averaged over four cameras and four persons, obtained from three different DJPDA algorithms are depicted in Fig. 6.8. It can be noted from this figure that both DJPDA-CM and DJPDA-HC asymptotically converge to the optimal centralised solution. Although DJPDA-CI has better tracking accuracy than DJPDA-CM with small number of consensus iterations, it cannot recover the performance of centralised fusion. Furthermore, the experimental study also reveals that the proposed DJPDA-HC outperforms both DJPDA-CI and DJPDA-CM in terms of tracking accuracy. These results clearly validate the theoretical study shown in previous sections.

6.6 Conclusions

The problem of distributed multiple targets tracking over a sensor network is investigated in this chapter. The proposed algorithms utilise the baseline JPDA filter in conjunction with different fusion strategies, e.g., CM, CI and HC, via the average

consensus algorithm. A distributed node counting algorithm is also proposed to support the implementation of DJPDA-HC. Theoretical analysis reveals that the proposed DJPDA-HC filter asymptotically converges to the optimal centralised JPDA and also provides the benefits of preserving the consistency of local estimates by a novel hybrid fusion strategy. Extensive simulations demonstrate that the proposed DJPDA-HC outperforms other algorithms under various different conditions.

Chapter 7

Practical Implementation of Multi-Dimensional Assignment

7.1 Introduction

It is known that solving the MDA problem is the key for successful implementing both MHT and multi-sensor MTT fusion. More specifically, the MDA is utilised in MHT to associate measurements to tracks to discriminate target-generated measurements and false alarms. Multi-sensor MTT requires formulating a MDA problem for track-to-track association to find the source origin of local estimates that come from different sensors. Conceptually, formulating MDA problem in these two different association scenarios is almost the same, except for the definition of the track hypothesis.

The main challenge in solving the MDA problem with problem size¹ larger than two is that the resulting problem is NP-hard. Due to this intrinsic property, finding optimal solution of the MDA problem is intractable even for fairly small-sized problems. The mainstream suboptimal solution of the MDA in both measurement-to-track and track-to-track associations is the well-established Lagrangian relaxation approach [65, 127, 66]. The basic idea of Lagrangian relaxation is to iteratively relax the original MDA problem to a 2D assignment problem, which can be efficiently solve by the well-known Hungarian algorithm. However, this algorithm too can become inefficient for fairly large scenarios, since it involves an iterative solution refinement

¹The problem size of MDA is defined as the number of scans in MHT or the number of sensors in multi-sensor MTT fusion.

procedure to produce better solutions to the original MDA problem. Therefore, it is imperative and meaningful to find an alternative solution to the MDA problem.

This chapter investigates the problem of practical implementation of MDA with performance-guaranteed approximation. The contribution of this chapter is the development of two different implementation algorithms for the MDA to mitigate the computational efficiency issue:

- The first approach takes advantages of the MWIS formulation and develops a meta-heuristic Tabu search algorithm to find the best global hypothesis for the MWIS problem formulated. The Tabu search based MDA implementation algorithm is also a new approach for the MWIS problem. The Tabu search is chosen because the MWIS problem is a combinatorial optimisation problem and the heuristic approach is considered to be well-suited to this type of problem [133, 134, 135]. At each Tabu iteration, two basic movement, namely add and swap, are leveraged to find a better neighbourhood solution of the initial candidate. A trigger policy is also proposed in the Tabu search to enhance the capability of escaping from the local optimum.
- The second approach proposed exploits countable approximation based on a stochastic sampling method. In this approach, we reformulate the max-summation MWIS problem to a new max-product decoding (MPD) problem. The key idea of the newly proposed formulation is to consider each solution of MWIS as a random variable with a stationary distribution. The product in the MPD problem is represented by this distribution and corresponds to the weight of MWIS. This chapter suggests to adopt the stochastic Gibbs sampling to approximate the distribution. It is then proven that the Markov chain generated by the proposed Gibbs sampling asymptotically converges to the target invariant distribution. This means the approximation error has theoretical upper bound, hence with guaranteed performance.

7.2 Preliminaries

This section introduces some basic concepts of MDA. We first formulate the MDA problems for both measurement-to-track association in MHT and track-to-track association and multi-sensor MTT. Then, the hypothesis pruning technique will be discussed.

7.2.1 MDA in Measurement-to-Track Association

The measurement-to-track association in MHT aims to find the most probable global hypothesis and use the matched state estimation from this particular hypothesis for the track maintenance. A global hypothesis is a set of tracks that are compatible, that is not in conflict: any two tracks in a global hypothesis do not share any common non-dummy observations at any scan and the two tracks cannot originate from one common node at the root scan. Assume the set of received measurements at scan k is modelled as Eq. (4.1) and S scans of measurements are utilised in data association. A track hypothesis² at scan k is defined as a sequence of measurements as

$$\mathcal{T}_k^i = \{\mathbf{z}_{k-S+1}^{i_1}, \mathbf{z}_{k-S+2}^{i_2}, \dots, \mathbf{z}_k^{i_S}\}, \quad \mathbf{z}_k^{i_S} \in \mathcal{Z}_k \quad (7.1)$$

Hypothesis test is usually leveraged to differentiate between target-generated measurements and false alarms in associating measurements to track hypothesis $\mathcal{T}_k^i$. Define $\mathcal{H}_1$ as the hypothesis that all measurements in $\mathcal{T}_k^i$ originate from the same target and $\mathcal{H}_0$ as the hypothesis that all measurements in $\mathcal{T}_k^i$ are false alarms. Under Gaussian assumption, the likelihood ratio³ between $\mathcal{H}_1$ and $\mathcal{H}_0$ is given by [136]

$$\begin{aligned} \Lambda(\mathcal{T}_k^i) &= \frac{p(\mathcal{T}_k^i | \mathcal{H}_1)}{p(\mathcal{T}_k^i | \mathcal{H}_0)} \\ &= \left[\prod_{s \in \mathcal{S}_m} \frac{P_D \mathcal{N}(\mathbf{z}_k^j; \mathbf{H}_k^i \mathbf{x}_k^i, \mathbf{S}_k^i)}{\lambda_{FA}} \right] \left[\prod_{s \notin \mathcal{S}_m} (1 - P_D) \right] \end{aligned} \quad (7.2)$$

where $\mathcal{S}_m \triangleq \{s_1, s_2, \dots, s_m\}$ with $s_i \in \{k - S + 1, k - S + 2, \dots, k\}$ is the set that contains scans that provide non-dummy measurements in data association, $\mathbf{H}_k^i$ the measurement matrix, and $\mathbf{S}_k^i$ the covariance of innovation.

Let $N_{\mathcal{T}}$ denote the total number of all possible track hypotheses during S scans, and M denote the summation of the number of resolved tracks at the root node, i.e., the confirmed tracks at the root scan, and the number of non-dummy measurements during the last $S - 1$ scans. Define $\mathbf{x} = [x_1, x_2, \dots, x_{N_{\mathcal{T}}}]^T$ as a binary vector with $x_i \in \{0, 1\}$. Then, the measurement-to-track association in MHT can be formulated

²Note that the size of one track hypothesis can be different from others due to new birth targets and death targets.

³The rationale of the utilisation of likelihood ratio is that it is a dimensionless quantity. Note that the likelihood functions have physical dimensions and therefore it is impossible to directly compare them.

as a constrained combinatorial optimisation problem, known as MDA, as [127]

$$\begin{aligned} & \arg \max_{\mathbf{x}} \mathbf{C}^T \mathbf{x} \\ & s.t. \quad \mathbf{A} \mathbf{x} = \mathbf{b}, \quad b_j = 1, \quad j = 1, 2, \dots, M \end{aligned} \quad (7.3)$$

where $\mathbf{C} = [\log \Lambda(\mathcal{T}_k^1), \log \Lambda(\mathcal{T}_k^2), \dots, \log \Lambda(\mathcal{T}_k^{N_{\mathcal{T}}})]^T$ is a vector of track hypothesis log-likelihood. Here, $x_i \in \{0, 1\}$ is a binary variable indicating the inclusion of the i th track hypothesis in the best global hypothesis with 1 for being included and 0 for being excluded. Matrix $\mathbf{A}$ is a $M \times N_{\mathcal{T}}$ is a binary matrix that is used to ensure that the solution $\mathbf{x}$ provides a feasible global hypothesis.

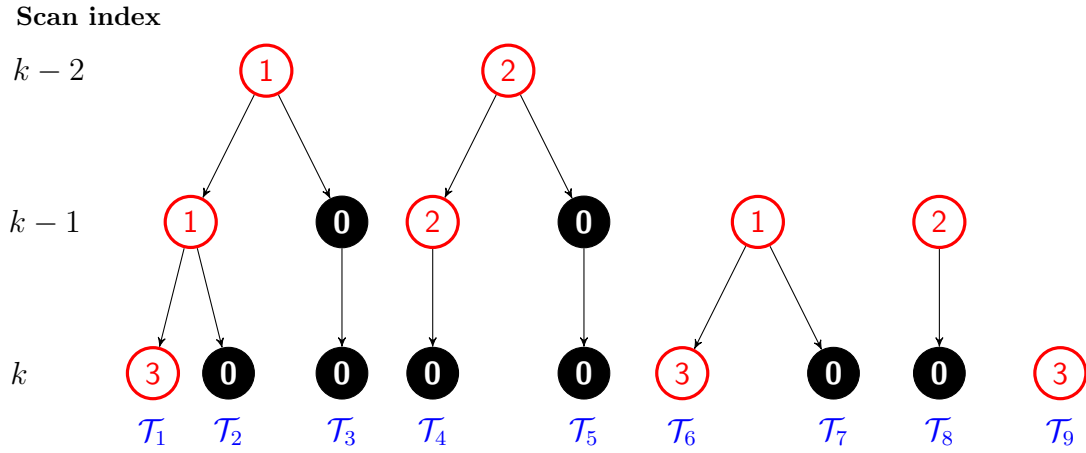


Figure 7.1: An example of track hypothesis tree for MDA formulation.

An example of track hypothesis tree for the MDA formulation with a three scan depth is shown in Fig. 7.1, where the numbers represent the measurement IDs that are associated with tracks. In this figure, the black circle refers to the dummy measurement. In this example, we assume that we have two resolved/confirmed tracks at scan $k-2$. At scan $k-1$ and k , the received measurement sets are $\{\mathbf{z}_{k-1}^1, \mathbf{z}_{k-1}^2\}$ and $\{\mathbf{z}_{k-2}^3\}$, respectively. When receiving a measurement, we create a new track hypothesis representing that the possibility that the measurement comes from a new target. As can be noted from Fig. 7.1, it can be noted that:

- Track hypotheses 1, 2 and 3 originate from the common node at the root scan $k-2$. This means that a valid global hypothesis should satisfy $x_1 + x_2 + x_3 = 1$.
- Track hypotheses 4 and 5 originate from the common node at the root scan $k-2$. This generates the constraint $x_4 + x_5 = 1$.
- Track hypotheses 1, 2, 6 and 7 share a common non-dummy observation at

scan $k - 1$ and therefore the solution should satisfy $x_1 + x_2 + x_6 + x_7 = 1$.

- Track hypotheses 4 and 8 share a common non-dummy observation at scan $k - 1$. This means that these two tracks are incompatible, i.e., $x_4 + x_8 = 1$.
- Track hypotheses 1, 6 and 9 share a common non-dummy observation at scan k . This means that a feasible global hypothesis is constrained by $x_1 + x_6 + x_9 = 1$.

Based these observations, the matrix $\mathbf{A}$ in Eq. (7.3) for this example is given by

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad (7.4)$$

where the first two rows are used to satisfy the constraint that any two tracks in a global hypothesis cannot originate from the same root node and the rest three rows are utilised to guarantee that one measurement can only be assigned to one track.

7.2.2 MDA in Track-to-Track Association

The track-to-track association in multi-sensor MTT aims to find the most probable global hypothesis and use the matched local estimates from this particular global hypothesis for the track fusion. A global hypothesis is a set of track hypotheses that are compatible, that is not in conflict: any two track hypotheses in a global hypothesis do not share any common non-dummy local estimate and the any two local estimates from one track hypothesis cannot have the same sensor index. Assume that N sensors participate in a cooperative estimation mission. Denote $\mathbf{x}_{l,k|k}^i$ as the local estimated state of the i th target provided by sensor l at scan k . A track hypothesis is defined as the set of local estimates as

$$\mathcal{T}_k^i = \left\{ \mathbf{x}_{1,k|k}^{i_1}, \mathbf{x}_{2,k|k}^{i_2}, \dots, \mathbf{x}_{N,k|k}^{i_N} \right\} \quad (7.5)$$

In order to adjust the problem that some sensors may miss detect some targets due to the non-unity detection probability, we define $i_l = 0$ as the index of the 'dummy' local estimate of the l th sensor. To differentiate between common and disperate origin of track hypothesis $\mathcal{T}_k^i$, we utilise the likelihood ratio $\Lambda(\mathcal{T}_k^i)$ between the hypothesis $\mathcal{H}_1$ that local estimates $i_1, i_2, \dots, i_N$ come from the same target and the hypothesis $\mathcal{H}_0$ that local estimates $i_1, i_2, \dots, i_N$ all have different origins. Denote

$\mathcal{L}_m \triangleq \{l_1, l_2, \dots, l_m\}$ with $l_i \in \{1, 2, \dots, N\}$ as the sensor set that contains nodes providing non-dummy local estimates. The likelihood ratio $\Lambda(\mathcal{T}_k^i)$ is given by⁴ [137]

$$\begin{aligned} \Lambda(\mathcal{T}_k^i) &= \frac{p(\mathcal{T}_k^i | \mathcal{H}_1)}{p(\mathcal{T}_k^i | \mathcal{H}_0)} \\ &= V^{m-1} \mathcal{N}(\mathbf{x}_{\mathcal{L}_m}^i; \mathbf{0}_{mn}, \mathbf{P}_{\mathcal{L}_m}^i) \left[\prod_{l \in \mathcal{L}_m} P_{D_l} \right] \left[\prod_{l \notin \mathcal{L}_m} (1 - P_{D_l}) \right] \end{aligned} \quad (7.6)$$

where V is the volume of the space; P_{D_l} denotes the detection probability of the l th sensor; and

$$\mathbf{x}_{\mathcal{L}_m}^i = \begin{bmatrix} \mathbf{x}_{l_2, k|k}^{i_{l_2}} - \mathbf{x}_{l_1, k|k}^{i_{l_1}} \\ \mathbf{x}_{l_3, k|k}^{i_{l_3}} - \mathbf{x}_{l_1, k|k}^{i_{l_1}} \\ \vdots \\ \mathbf{x}_{l_m, k|k}^{i_{l_m}} - \mathbf{x}_{l_1, k|k}^{i_{l_1}} \end{bmatrix} \in \mathbb{R}^{mn} \quad (7.7)$$

is a stacked vector, whose covariance $\mathbf{P}_{\mathcal{L}_m}^i$ has the diagonal blocks

$$(\mathbf{P}_{\mathcal{L}_m}^i)_{s-1, s-1} = \mathbf{P}_{l_1, k|k}^{i_{l_1}} + \mathbf{P}_{l_s, k|k}^{i_{l_s}} - \mathbf{P}_{l_1, l_s, k|k}^{i_{l_1}, i_{l_s}} - \left(\mathbf{P}_{l_1, l_s, k|k}^{i_{l_1}, i_{l_s}} \right)^T, \quad s = 2, 3, \dots, m \quad (7.8)$$

and the off diagonal blocks

$$(\mathbf{P}_{\mathcal{L}_m}^i)_{s-1, s'-1} = \mathbf{P}_{l_1, k|k}^{i_{l_1}} - \mathbf{P}_{l_1, l_{s'}, k|k}^{i_{l_1}, i_{l_{s'}}} - \left(\mathbf{P}_{l_1, l_s, k|k}^{i_{l_1}, i_{l_s}} \right)^T + \mathbf{P}_{l_s, l_{s'}, k|k}^{i_{l_s}, i_{l_{s'}}}, \quad s, s' = 2, 3, \dots, m \quad (7.9)$$

with $\mathbf{P}_{l_s, l_{s'}, k|k}^{i_{l_s}, i_{l_{s'}}}$ being the cross-covariance of two local estimates that from two different sensors, resulting from the common process noise, and can be obtained by a recursive update or the approximated approach shown in [137].

Similar to measurement-to-track association, the track-to-track association can also be formulated as a MDA problem as [69, 138]

$$\begin{aligned} &\arg \max_{\mathbf{x}} \mathbf{C}^T \mathbf{x} \\ \text{s.t. } &\mathbf{A} \mathbf{x} = \mathbf{b}, \quad b_j = 1, \quad j = 1, 2, \dots, M \end{aligned} \quad (7.10)$$

where M is the summation of the number of local estimates of the reference sensor and the number of non-dummy local estimates of the rest $N - 1$ sensors. Other parameters share the same definitions as Eq. (7.3).

⁴Note that we utilise the track from the l_1 th sensor as the reference sensor in associating local estimated tracks. It is obviously legitimate that any other sensors can be used as the reference.

7.2.3 Hypothesis Pruning

Ideally, all the candidate track hypotheses need to be considered in the solution space. The issue is that the number of track hypotheses exponentially increases with the increase of the problem size. This implies that considering all the tracks may become intractable even with heuristic or approximated approaches. One most commonly accepted approach to handle this issue is hypothesis-level pruning, which removes track hypotheses with low probability. Hypothesis-level pruning can easily be performed using the approximated track hypothesis probability (ATHP) [50]. Based on the law of total probability, the track hypothesis probability $p(\mathcal{T}_k^i)$ can be calculated as

$$p(\mathcal{T}_k^i) = \sum_{\mathcal{H}_k \ni \mathcal{T}_k^i} p(\mathcal{H}_k) \quad (7.11)$$

where $p(\mathcal{H}_k)$ denotes the probability of the global hypothesis $\mathcal{H}_k$, and is defined as

$$p(\mathcal{H}_k) = \frac{\prod_{\mathcal{T}_k^i \in \mathcal{H}_k} \lambda(\mathcal{T}_k^i)}{1 + \sum_{\mathcal{H}_k} \prod_{\mathcal{T}_k^i \in \mathcal{H}_k} \lambda(\mathcal{T}_k^i)} \quad (7.12)$$

Any track hypothesis with probability that is below than a lower threshold p_{th} is removed. Applying the hypothesis-level pruning, poor track hypotheses can be removed and consequently only high-quality tracks are considered in association.

7.3 MWIS Formulation

The best global hypothesis generation problem in MDA can be alternatively formulated as an MWIS problem of a weighted undirected graph. This problem reformulation is the foundation of the development of the proposed algorithms.

An independent set of an undirected graph is defined as a subset with no pairwise adjacent elements of the vertex set. The MWIS is the heaviest subset of non-adjacent nodes/vertices of a graph. A weighted undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ at each scan for this problem is constructed by assigning a track hypothesis $\mathcal{T}_i^k$ to a graph vertex v_i and each vertex with a weight w_i corresponds to its log-likelihood ratio $\log \lambda(\mathcal{T}_i^k)$. If two track hypotheses cannot co-exist, i.e., they are incompatible, an edge is constructed to connect them. Let $|\mathcal{V}|$ denote the cardinality, i.e., the number of elements, of vertex set $\mathcal{V}$. Then, the problem of best global hypothesis

generation in MDA can be transformed to finding the MWIS as

$$\begin{aligned} \arg \max_{\mathbf{x}} \sum_i^{|\mathcal{V}|} w_i x_i \\ s.t. \quad x_i + x_j \leq 1, \quad \forall (v_i, v_j) \in \mathcal{E} \end{aligned} \quad (7.13)$$

where $\mathbf{x} \triangleq [x_1, x_2, \dots, x_{|\mathcal{V}|}]^T$ and $x_i \in \{0, 1\}$ is a binary variable that indicates whether or not the i th track hypothesis is included in the solution.

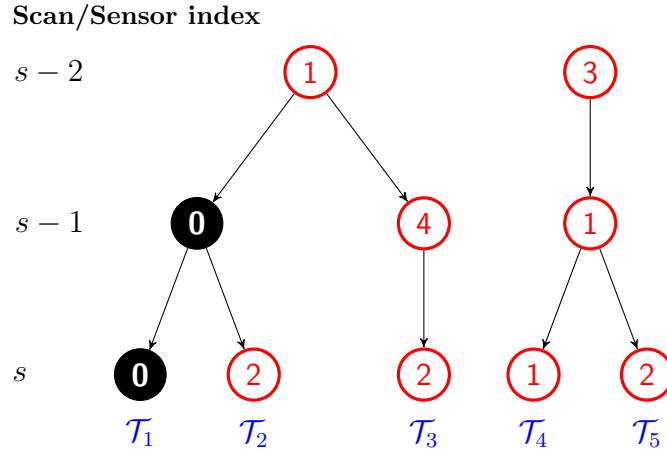


Figure 7.2: Hypothesis history of the considered scenario.

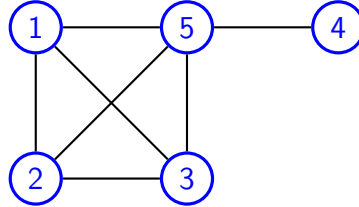


Figure 7.3: Undirected graph of the considered example.

Here, we provide a simple example to show how to transform the MDA problem into a MWIS problem. The hypothesis history of the considered scenario is illustrated in Fig. 7.2, where the term 'sensor' is utilised in track-to-track association and 'scan' is used in measurement-to-track association. The number in the circle represents the ID of local estimates (in track-to-track association) or measurements (in measurement-to-track association). In this figure, the black circle refers to the dummy measurement/local estimate. For this simple example, track hypotheses 1, 2 and 3 originate from a common parent node 1; track hypotheses 4 and 5 originate from a common node 3; track hypotheses 2, 3 and 5 share a common non-dummy

node, i.e., node 2. Based these observations, one can construct an undirected graph as shown Fig. 7.3, where the numbers are IDs of track hypotheses.

7.4 Solving MDA Problem Using Heuristic Tabu Search

This section proposes a Tabu search based approach to solve the MDA problem. Tabu search is one of the state-of-the-art general metaheuristic algorithms to find the approximate optimal solutions for a wide variety of hard combinatorial optimisation problems [139]. Tabu search usually consists of two phases: initial solution generation phase and local search phase. Compared with other heuristic algorithms, Tabu search enhances the performance of local search by introducing prohibition rules, which prevent the search from visiting the previously-searched candidate solutions.

For any feasible solution $\mathcal{F} = \{i \in \mathcal{V} : i \neq j \forall j \in \mathcal{V} \text{ s.t. } \{i, j\} \notin \mathcal{E}\}$, the quality of the solution, denoted as $W(\mathcal{F}) = \sum_i^{|\mathcal{V}|} w_i x_i$, is quantified by the total weight (summation) of all vertices. Given two solutions $\mathcal{F}_1$ and $\mathcal{F}_2$, $\mathcal{F}_1$ is better than $\mathcal{F}_2$ if and only if $W(\mathcal{F}_1) > W(\mathcal{F}_2)$. The object function of Tabu search is thus given by $W(\mathcal{F})$. The aim here is to find $\mathcal{F}$ maximising $W(\mathcal{F})$ using two phases, i.e., initial phase and local search phase.

7.4.1 Initial Solution Generation

The goal of the initial phase is to randomly generate a set of possible candidate solutions. To achieve this, we first sort all tracks in a descending manner based on their track scores. The initial phase starts from an initial solution $\mathcal{F}$ by randomly selecting a unique track vertex from the first N_1 sorted tracks as these N_1 tracks have relatively higher scores. Then, iteratively, randomly pick up a compatible track vertex $v = \{v : v \in \mathcal{V} \setminus \mathcal{F}, (v, l) \notin \mathcal{E}, \forall l \in \mathcal{F}\}$ and add it into $\mathcal{F}$ until there is no feasible vertex can be added based on the independence constraint. In general, the initial phase only aims to find a candidate solution, i.e., a global hypothesis, and ignores the quality of the solution, but ensures a rich diversity of the initial solutions for local search in a different region. This is achieved by using a search length strategy that triggers a new initial solution if the quality of the previous solution never improves for a number of consecutive iterations.

7.4.2 Local Search Phase

Local search is used to provide an attempt to find better immediate neighbourhood solutions for the initial solutions. A neighbourhood of a candidate solution confines the search space and is typically defined by a move operator *move*. This operator transforms a given solution $\mathcal{F}$ to a better neighbourhood solution, that is $\bar{\mathcal{F}} \triangleq \mathcal{F} \oplus \text{move}$. The proposed algorithm considers two natural operators: $ADD(i)$ and $SWAP(i, j)$. These two operators are defined based on two sets: $\mathcal{M}_A$ and $\mathcal{M}_S$. The set $\mathcal{M}_A$ is composed of the vertices that are all excluded from the candidate solution $\mathcal{F}$ and not adjacent to all vertices of $\mathcal{F}$. The set $\mathcal{M}_S$ consists of the vertices that are all excluded from the set $\mathcal{F}$ and adjacent to only one vertex in $\mathcal{F}$. The operator $ADD(i)$ is defined as adding a vertex in $\mathcal{M}_A$ to $\mathcal{F}$. The neighbourhood solution resulted from this operator is $\bar{\mathcal{F}}_A = \{\bar{\mathcal{F}}_A : \mathcal{F} \oplus ADD(i), i \in \mathcal{M}_A\}$. Clearly, the $ADD(i)$ move operator always improves the quality of a solution. The operator $SWAP(i, j)$ is defined as replacing an existed vertex $j \in \mathcal{F}$ with its adjacent vertex $i \in \mathcal{M}_S$ to generate a neighbourhood solution: $\bar{\mathcal{F}}_S = \{\bar{\mathcal{F}}_S : \mathcal{F} \oplus SWAP(i, j), i \in \mathcal{M}_S\}$. This operator can either improve or deteriorate the quality of a solution. From this analysis, it is clear that an overall better solution can be selected by a union of these two neighbourhood solutions, i.e., a hybrid movement. Note that both $\mathcal{M}_A$ and $\mathcal{M}_S$ could be an empty set if there is no such set exists.

Tabu search introduces the so-called Tabu list to discourage the search from coming back to previously-examined solutions. Invoking the fact that the operator $ADD(i)$ always improves the quality of the current solution, the prohibition rule in the proposed algorithm is designed as: a vertex that joins the solution $\mathcal{F}$ is free to leave $\mathcal{F}$ while a vertex that is removed from the solution is not allowed to be added again into $\mathcal{F}$ for next T_S iterations (Tabu tenure) except for the case that it provides a better solution.

Consider the undirected graph shown in Fig. 7.4, where the node weights are the same as the IDs. Suppose the candidate solution is given by $\mathcal{F} = \{1, 4, 5, 7\}$. For $ADD(i)$, it is easy to verify that $\mathcal{M}_A$ is an empty set based on the independence constraint. Similarly, one can obtain $\mathcal{M}_S = \{6, 8, 9\}$ for $SWAP(i, j)$ and thus replacing $\{5\}$ with $\{6\}$ and replacing $\{7\}$ with $\{8\}$ or $\{9\}$ result in a better solution. Since the weight of node 9 is larger than that of node 6 and 8, the local search automatically chooses node 9 at the current iteration, leading to an improved candidate solution $\mathcal{F} = \{1, 4, 5, 9\}$. In the next iteration, one can also obtain $\mathcal{M}_A = \{8\}$

for $ADD(i)$ and $\mathcal{M}_S = \{6, 7\}$ for $SWAP(i, j)$. Since node 7 is excluded in the previous update step and thus will be automatically blocked by the local search phase through a Tabu list as discussed before. Consequently, the most favourable solution in terms of its quality obtained by these two operators at the this iteration is $\mathcal{F} = \{1, 4, 6, 8, 9\}$.

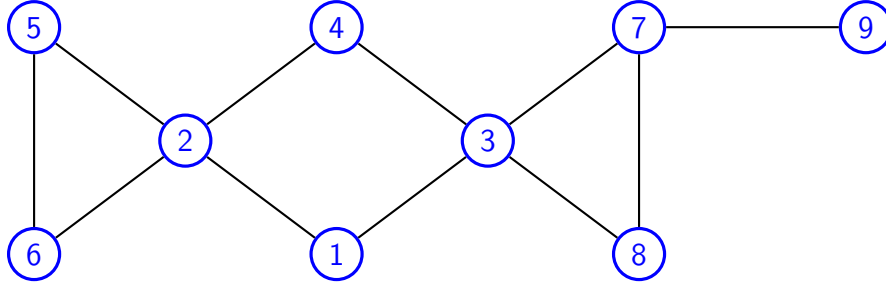


Figure 7.4: Undirected graph of the considered example.

The proposed algorithm examines the possible neighbourhood solutions based on their qualities at every iteration and selects the most preferable one. The prohibition rule provides the possibility of local diversification by preventing the previously-visited solutions from being re-searched for next several iterations. A new restart from a different randomised initial solution is triggered if the quality of the solution obtained by the local search never improves for a maximum allowable consecutive search depth. Although the heuristic approach cannot have theoretical guarantee of the optimal solution, the proposed randomised multi-restart policy can help the search to escape from the local basin and hence can help to prevent from being trapped in a local optimum solution.

Track hypothesis level pruning using the ATHP concept is performed to enhance the computational efficiency. For pruning, we first define a set $\mathcal{B}_{global}$ that is composed of feasible global hypothesis candidates and update it at every iteration. Using global hypotheses in the set $\mathcal{B}_{global}$ obtained, the proposed algorithm then calculates the ATHP for each track, whose equation is given in Eq. (7.11), and prunes the ones smaller than the threshold value designed.

The algorithm of the proposed Tabu search-based best global hypothesis generation is summarised in Algorithm 7.1.

Algorithm 7.1 Best global hypothesis generation by Tabu search

Input: A weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ generated by all tracks at scan k , maximum allowable search depth L , maximum allowable iterations n_{max}

Output: The best global hypothesis defined by set $\mathcal{F}^*$, global hypothesis candidate set $\mathcal{B}_{global}$

```

1:  $iter \leftarrow 0$  ▷ Set the initial iteration counter
2:  $\mathcal{F}^* \leftarrow \emptyset$  ▷ Set the initial solution
3:  $\mathcal{B}_{global} \leftarrow \emptyset$  ▷ Set the initial global hypothesis candidate
4: while  $iter < n_{max}$  do
5:    $\mathcal{F}^* \leftarrow$  random initial possible solution generated from  $\mathcal{G}$ 
6:   Initialise the Tabu list
7:    $\mathcal{F}_{local} \leftarrow \mathcal{F}$  ▷  $\mathcal{F}_{local}$  is the local optimal solution in each iteration
8:    $L_{local} \leftarrow 0$  ▷ Initialise the local search step
9:   while  $L_{local} < L$  do
10:     $\bar{\mathcal{F}}_A, \bar{\mathcal{F}}_S \leftarrow$  construct neighbourhood solutions based on  $ADD(i)$  and  $SWAP(i, j)$ 
11:     $\mathcal{F} \leftarrow$  select the overall best solution  $\bar{\mathcal{F}}_A \cup \bar{\mathcal{F}}_S$  according to the quality function
12:     $L_{local} \leftarrow L_{local} + 1$ 
13:     $iter \leftarrow iter + 1$ 
14:    Update the Tabu list
15:    if  $W(\mathcal{F}) > W(\mathcal{F}_{local})$  then
16:       $L_{local} \leftarrow 0$ 
17:       $\mathcal{F}_{local} \leftarrow \mathcal{F}$ 
18:    end if
19:  end while
20:  if  $\mathcal{F}_{local} \notin \mathcal{B}_{global}$  then
21:     $\mathcal{B}_{global} \leftarrow \{\mathcal{B}_{global}, \mathcal{F}_{local}\}$  ▷ Update the global hypothesis candidate set
22:  end if
23:  if  $W(\mathcal{F}_{local}) > W(\mathcal{F}^*)$  then
24:     $\mathcal{F}^* \leftarrow \mathcal{F}_{local}$  ▷ Update the best global hypothesis
25:  end if
26: end while
27: return  $\mathcal{F}^*, \mathcal{B}_{global}$ 

```

7.5 Solving MDA Problem Using Stochastic Gibbs Sampling

This section proposes an alternative approximated solution of the original MWIS problem described in (7.13). Our proposition is to apply Gibbs sampling to sample candidate solutions of x in (7.13) and evaluate their corresponding quality. As candidate solutions and their corresponding quality are recursively generated, sorting the solutions and finding the approximated solution, providing the best quality, are straightforward.

To apply Gibbs sampling, it is required to reformulate the original MWIS problem into a problem in which the key idea of Gibbs sampling can be applied. Therefore, we first reformulate the original max-sum problem given in (7.13) to a new max-product problem based on the property of log-likelihood ratio. For this reason, define

$$p(\mathbf{x}) \triangleq \left(\prod_{i \in \mathcal{V}} \varphi(x_i) \right) \left(\prod_{(v_i, v_j) \in \mathcal{E}} \varphi_c(x_i, x_j) \right) \quad (7.14)$$

with

$$\varphi(x_i) = \begin{cases} e^{\lambda(\mathcal{T}_k^i)}, & x_i = 1 \\ 1, & x_i = 0 \end{cases} \quad (7.15)$$

$$\varphi_c(x_i, x_j) = \begin{cases} 0, & x_i = x_j = 1 \\ 1, & \text{otherwise} \end{cases} \quad (7.16)$$

where $\varphi(x_i)$ is the node potential, guaranteeing a non-negative un-normalised probability to each possible value of the random variable $x_i \in \{0, 1\}$; $\varphi_c(x_i, x_j)$ is the edge potential, providing a non-negative un-normalised weight to all the combinations that (x_i, x_j) can take. The function $p(\mathbf{x})$ can be regarded as an un-normalised joint distribution of a pairwise undirected graphical model.

Then, the original MWIS problem can be equivalently solved by the following optimisation problem

$$\arg \max_{\mathbf{x}} p(\mathbf{x}) \quad (7.17)$$

The solution of (7.17) is a decoding problem, in which the goal is to find the most probable joint distribution. The proposed max-product formulation enables utilisation of Gibbs sampling. The key idea proposed in this chapter is to consider each solution $\mathbf{x}$ as a random variable that satisfies a distribution $\pi(\mathbf{x})$. For the realisation, we first construct a finite-state Markov chain whose state space is the set of

all feasible solutions (i.e. independent sets of graph $\mathcal{G}$). Then, the Gibbs sampler is utilised to generate random solution samples $\mathbf{x}$ based on the proposed distribution proposal $\pi(\mathbf{x})$ so as to find the best candidate set. To ensure that the independent set solution with higher weights are more likely sampled, it is natural to construct the sampling proposal π proportional to its corresponding weight/track scores as

$$\pi(\mathbf{x}) \propto \left(\prod_{i=1}^{|\mathcal{V}|} \varphi(x_i) \right) \left(\prod_{(v_i, v_j) \in \mathcal{E}} \varphi_c(x_i, x_j) \right) \quad (7.18)$$

Enumerating all possible independent sets is impossible for real applications, which means direct sampling from (7.18) is difficult. Hence, the concept of Gibbs sampling is adopted here, i.e. samples are generated by sweeping through all the posterior conditionals, rather than from the posterior distribution itself. Then, the one-step transition kernel from one possible solution $\mathbf{x} = [x_1, \dots, x_{|\mathcal{V}|}]^T$ to another possible solution $\bar{\mathbf{x}} = [\bar{x}_1, \dots, \bar{x}_{|\mathcal{V}|}]^T$ is obtained as

$$\pi(\bar{\mathbf{x}}|\mathbf{x}) = \prod_{m=1}^{|\mathcal{V}|} \pi_m(\bar{x}_m | \bar{x}_1, \dots, \bar{x}_{m-1}, x_{m+1}, \dots, x_{|\mathcal{V}|}) \quad (7.19)$$

where the condition distribution is given by

$$\pi_m(\bar{x}_m | \bar{x}_1, \dots, \bar{x}_{m-1}, x_{m+1}, \dots, x_{|\mathcal{V}|}) \propto \varphi(\bar{x}_i) \quad (7.20)$$

Given a solution $\mathbf{x}$, another feasible solution $\bar{\mathbf{x}}$ can then be obtained by recursive sampling according to the following individual conditional distributions

$$\begin{aligned} \bar{x}_1 &\sim \pi_1(\bar{x}_1 | x_2, \dots, x_{|\mathcal{V}|}) \\ &\vdots \\ \bar{x}_m &\sim \pi_m(\bar{x}_m | \bar{x}_1, \dots, \bar{x}_{m-1}, x_{m+1}, \dots, x_{|\mathcal{V}|}) \\ &\vdots \\ \bar{x}_{|\mathcal{V}|} &\sim \pi_{|\mathcal{V}|}(\bar{x}_{|\mathcal{V}|} | \bar{x}_1, \dots, \bar{x}_{|\mathcal{V}|-1}) \end{aligned} \quad (7.21)$$

Similar to Theorem 4.1, it can be proved that the generated Markov chain asymptotically converges to its invariant distribution with an exponential convergence rate as

$$|\pi^n(\bar{\mathbf{x}}|\mathbf{x}) - \pi(\bar{\mathbf{x}})| \leq (1 - 2\beta)^{\lfloor n/2 \rfloor} \quad (7.22)$$

where $\pi^n(\bar{\mathbf{x}}|\mathbf{x})$ denotes the n th power of transition kernel $\pi(\bar{\mathbf{x}}|\mathbf{x})$, $\beta = \min_{\mathbf{x}, \bar{\mathbf{x}}} \pi^2(\bar{\mathbf{x}}|\mathbf{x}) \in$

$(0, 0.5]$ the least likely two-step transition probability.

Algorithm 7.2 summarises the proposed Gibbs sampling-based implementation algorithm.

Algorithm 7.2 Best global hypothesis generation by Gibbs sampling

Input: A weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ generated by all tracks at scan k , maximum allowable Gibbs samples n_{gibbs} , burn-in samples $n_{burn-in}$

Output: The best global hypothesis defined by $\mathbf{x}_{best}$, marginal distribution $p(x_i)$, $\forall i \in [|\mathcal{V}|]$

```

1:  $iter \leftarrow 0$  ▷ Set the initial iteration counter
2: Randomly picking an initial possible solution  $\mathbf{x}^{(1)}$ 
3:  $p_{best} \leftarrow -\inf$  ▷ Set the initial best cost as -inf
4: if  $i = 1 : |\mathcal{V}|$  then
5:      $n_i \leftarrow 0$  ▷ Set the initial event counter
6: end if
7: while  $iter < n_{max}$  do
8:     if  $i = 1 : |\mathcal{V}|$  then
9:         Recursive sampling according to (7.21)
10:    end if
11:     $\mathbf{x}^{(iter)} = [x_1^{(iter)}, \dots, x_{|\mathcal{V}|}^{(iter)}]^T$  ▷ One Gibbs sample
12:    if  $iter > n_{burn-in}$  then
13:        Evaluate the current solution  $p(\mathbf{x}^{(iter)})$ 
14:        if  $p(\mathbf{x}^{(iter)}) > p_{best}$  then
15:             $p_{best} \leftarrow p(\mathbf{x}^{(iter)})$ 
16:             $\mathbf{x}_{best} \leftarrow \mathbf{x}^{(iter)}$ 
17:        end if
18:        for  $i = 1 : |\mathcal{V}|$  do
19:            if  $x_i^{(iter)} = 1$  then
20:                 $n_i \leftarrow n_i + 1$ 
21:            end if
22:        end for
23:    end if
24:     $iter \leftarrow iter + 1$ 
25: end while
26: for  $i = 1 : |\mathcal{V}|$  do
27:      $p(x_i = 1) \leftarrow n_i / (n_{max} - n_{burn-in})$  ▷ Approximated marginal probability
28: end for
29: return  $\mathbf{x}_{best}, p(x_i)$ 

```

Remark 7.1. It follows from Eq. (7.22) that, given any initial feasible solution $\mathbf{x}^{(1)}$, the generated Markov chain $\{\mathbf{x}^{(t)}\}_{t=1}^{\infty}$ exponentially converges to its stationary invariant distribution. Due to the convergence property, one can easily select a feasible solution as an initial state for Gibbs sampler. For example, one can simply

choose the initial solution by random selection of compatible tracks, as presented in the previous section.

Remark 7.2. *With the aid of Gibbs sampler, one can also easily obtain the approximated marginal track hypothesis probability $p(x_i = 1)$ as*

$$p(x_i = 1) = \frac{n_i}{n_{\max} - n_{\text{burn-in}}} \quad (7.23)$$

where $n_{\max}$ denotes maximum allowable Gibbs samples, $n_{\text{burn-in}}$ the burn-in samples, n_i the number of samples that $x_i = 1$ occurs after the pre-convergent burn-in phase. Using Eq. (7.23), we can also perform hypothesis-level pruning to improve the computational efficiency by thresholding $p(x_i = 1)$.

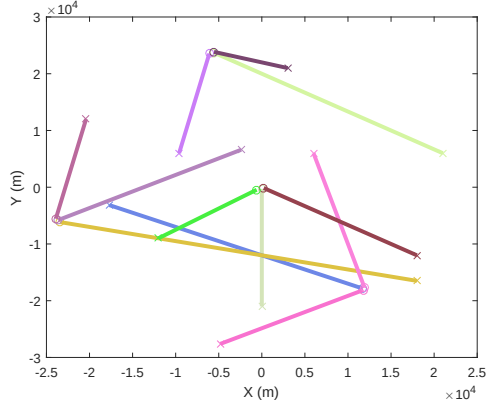
7.6 Numerical Simulations

This section validates the proposed algorithms through numerical simulations. In this section, the measurement-to-track association in MHT is utilised as an example to show the effectiveness of the proposed MDA implementation algorithms. To demonstrate the effectiveness, the performance of the proposed algorithms are compared with that of the mainstream MDA formulation of MHT with Lagrangian relaxation approach. As shown in [69], Lagrangian relaxation implementation algorithm outperforms both m -best assignment and linear programming solutions. Therefore, it is reasonable to compare the proposed methods with the Lagrangian approach.

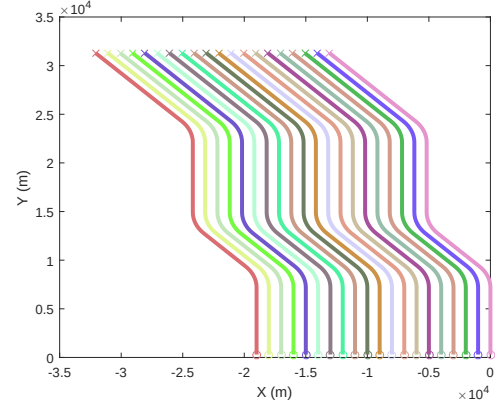
7.6.1 Simulation Setup

Our experiments six different challenging scenarios:

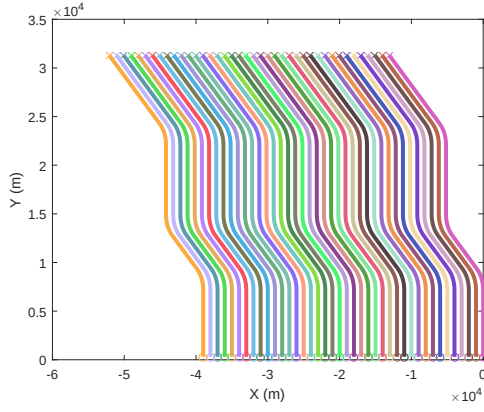
- Scenario 1: twelve constant moving targets with different birth and death time;
- Scenario 2: 20 close moving targets with three 3g coordinated turns;
- Scenario 3: 40 close moving targets with three 3g coordinated turns;
- Scenario 4: 20 random moving targets;
- Scenario 5: 40 random moving targets;
- Scenario 6: 60 random moving targets.



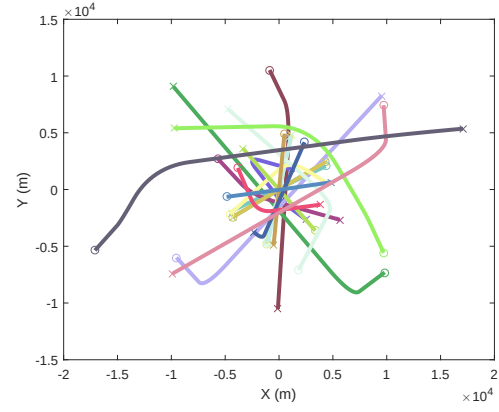
(a) Scenario 1



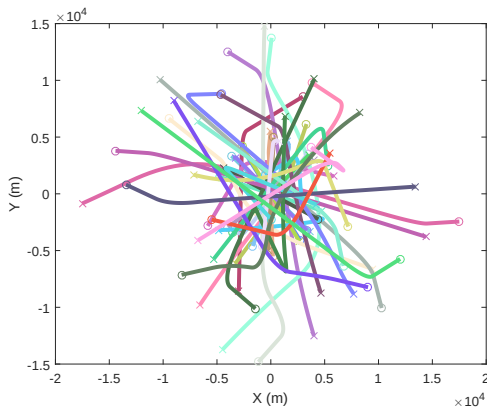
(b) Scenario 2



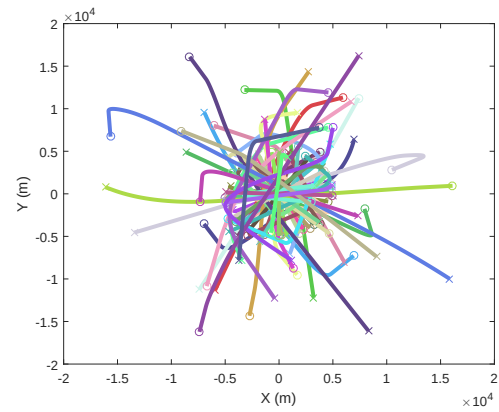
(c) Scenario 3



(d) Scenario 4



(e) Scenario 5



(f) Scenario 6

Figure 7.5: Sample ground truths of the considered scenarios.

The sample ground truths of these six scenarios are presented in Fig. 7.5. The state vector contains planar position and velocity. We assume that all targets share the same motion model and use the well-known CV model for target prediction. The CV model is defined as

$$\mathbf{x}_k^i = \mathbf{F}_{CV} \mathbf{x}_{k-1}^i + \mathbf{G} w_{k-1}^i \quad (7.24)$$

with

$$\mathbf{F}_{CV} \triangleq \mathbf{I}_2 \otimes \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}, \quad \mathbf{G} \triangleq \begin{bmatrix} T_s^2/2 \\ T_s \\ T_s^2/2 \\ T_s \end{bmatrix} \quad (7.25)$$

where $T_s = 1s$ is the sampling period, and $w_k^i \sim \mathcal{N}(\cdot; 0, \sigma_v^2)$ the Gaussian process noise with $\sigma_v = 22m/s^2$.

The sensor collects position measurements at regular time instants with detection probability $P_D = 0.95$ as

$$\mathbf{H}_k^i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (7.26)$$

The measurement noise is subject to a Gaussian white noise as $v_{k,i} \sim \mathcal{N}(\cdot; 0, \mathbf{R}_{k,i})$ with $\mathbf{R}_{k,i} = \text{diag}(\sigma_r^2, \sigma_r^2)$, $\sigma_r = 100m$. The clutter is assumed to be uniformly distributed in the surveillance region with its number at one scan being Poisson with spatial density $\lambda_{FA} = 10^{-7}$.

For every new track, the state estimation error covariance is initialised using the one-point method [57] with maximum velocity bound $600m/s$ and the new target density is set as $\lambda_B = 10^{-11}$. Since the detection probability and clutter spatial density may vary in practical implementations, we choose these two parameters as $P_D = 0.75$ and $\lambda_{FA} = 10^{-8}$, which are not matched with the ones used in measurement generation. For measure-to-track association, three scans of measurements are utilised in MHT. In order to remove the hypothesis with low probability, the threshold of hypothesis level pruning is selected as $p_{th} = 10^{-6}$. The design parameters for Tabu search based approach are chosen as $L = 10$, $n_{max} = 100$, and parameters for Gibbs sampling based implementation algorithm are set as $n_{max} = 2000$, $n_{burn-in} = 1000$.

The status of a track hypothesis in MHT is usually characterised by the classical sequential probability ratio test (SPRT) based on its current score, i.e., log likelihood hood ration $\log \Lambda(\mathcal{T}_k^i)$, versus the upper and lower thresholds (T_{low}, T_{up}) . Generally, a track is confirmed once its score exceeds the upper threshold T_{up} while a track is immediately deleted if its score is below the lower threshold T_{low} . If the track

score falls between the upper and lower thresholds, the corresponding track is still tentative and is required to be further tested. Following the standard SPRT, the upper and lower thresholds are defined as

$$T_{low} = \ln \left(\frac{\beta}{1 - \alpha} \right), \quad T_{up} = \ln \left(\frac{1 - \beta}{\alpha} \right) \quad (7.27)$$

where α is the false track confirmation probability, and β the true track deletion probability⁵. These two parameters are, respectively set as $\alpha = 10^{-6}$ and $\beta = 10^{-3}$

7.6.2 Performance Metric

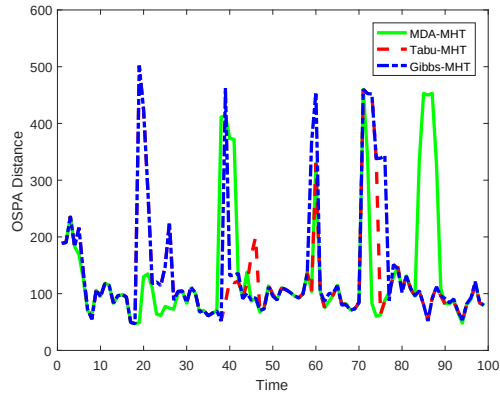
For performance evaluation, the following commonly accepted metrics [140] are considered.

- Mis correlation ratio of true tracks R_{MC} . The ratio of the number of false measurement-to-track associations to total true tracks;
- Correct correlation ratio of true tracks R_{CC} . The ratio of the number of correct measurement-to-track associations to total true tracks;
- Average number of true target IDs associate with a true track N_{ID} . For perfect matching in the ideal case, all measurements that allocated to a true track; should match to the true target, i.e. $N_{ID} = 1$. Similarly, $N_{ID} > 1$ holds for imperfect matching;
- Average true track life T_{avg} . Average true track life measured in scan numbers;
- Running time t_e . The execution time for a sample run;
- OSPA distance with $p = 2$, $c = 1000$.

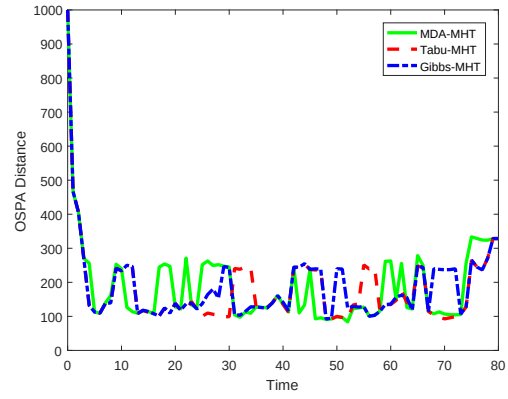
7.6.3 Simulation Results

Table 7.1 summarises all the performance metrics obtained by these three different approaches of fifty Monte-Carlo runs, where the values are calculated as the mean

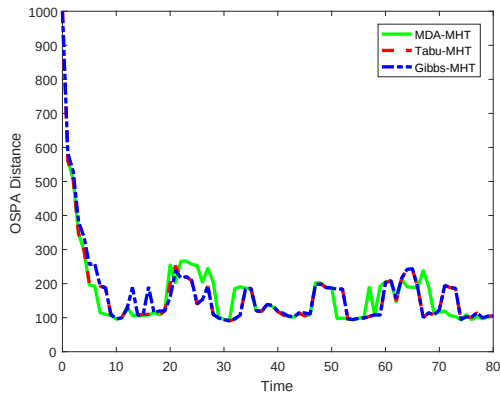
⁵The allowable false track confirmation probability α can be defined from the system requirements on false track initiation. For example, if the system generates N_{FA} false alarms per scan and only N_{FC} false track confirmations are allowed within S scans, then, one can set $\alpha = N_{FA}/(SN_{FC})$. Note that the choice of β is less important since the true track deletion probability β has less effect on the track confirmation threshold. Typical choice of β is $\beta \leq 0.1$.



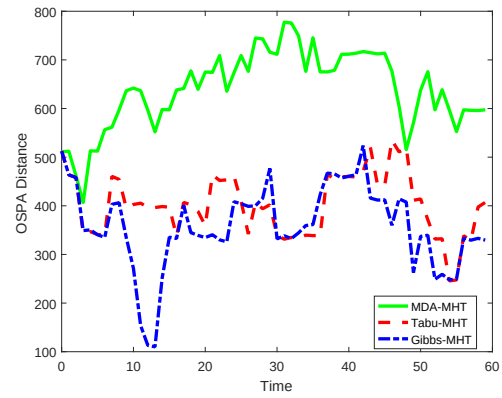
(a) Scenario 1



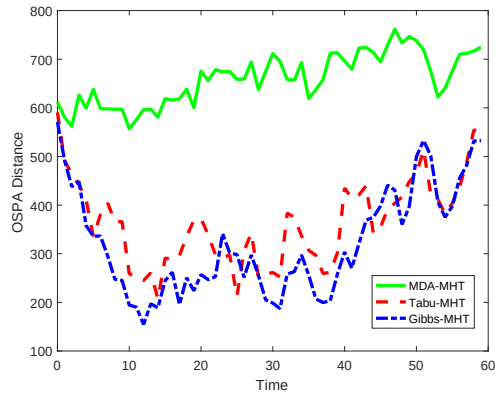
(b) Scenario 2



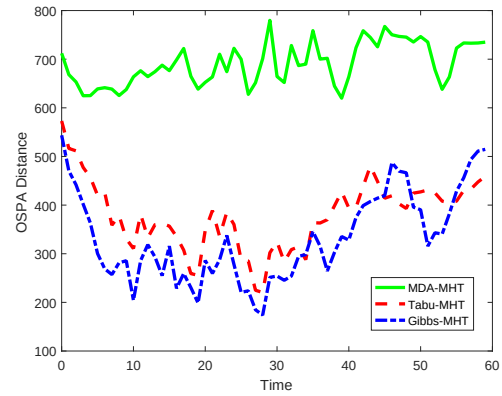
(c) Scenario 3



(d) Scenario 4



(e) Scenario 5



(f) Scenario 6

Figure 7.6: OSPA distance with MDA-MHT (green), Tabu-MHT (red) and Gibbs-MHT (blue).

Table 7.1: Comparison results of Monte-Carlo simulations.

Scenario	Tracker	R_{MC}	R_{CC}	N_{ID}	T_{avg}	t_e/s	OSPA
1	MDA	0.0671	0.9654	1.2265	44.2603	23.8042s	191.4025
	Tabu	0.0835	0.9636	1.3087	53.9755	20.2255s	179.9225
	Gibbs	0.0809	0.9660	1.1864	50.0071	16.5422	181.6081
2	MDA	0.0784	0.9538	1.0568	53.6069	7.7664s	153.5685
	Tabu	0.0882	0.9549	1.0604	63.4154	5.9839s	149.8836
	Gibbs	0.0854	0.9557	1.0470	53.6069	5.0561	152.0427
3	MDA	0.0758	0.9570	1.0414	51.2003	15.9088s	159.4479
	Tabu	0.0911	0.9540	1.0609	62.0841	13.1394s	154.6242
	Gibbs	0.0854	0.9591	1.0383	57.6434	11.3569	155.4897
4	MDA	0.2279	0.4102	2.3988	15.0391	11.3561s	597.8526
	Tabu	0.2386	0.4382	1.6182	24.9880	8.5660s	357.7429
	Gibbs	0.2479	0.4502	1.6652	26.4104	5.1800	317.1836
5	MDA	0.2142	0.3013	2.7500	11.6479	237.5427s	687.0442
	Tabu	0.2242	0.3313	2.8577	22.0750	37.1126s	362.0764
	Gibbs	0.2557	0.3160	2.7442	26.5581	19.6717s	323.1499
6	MDA	0.2138	0.3042	2.9515	12.3592	1403.6657s	786.1761
	Tabu	0.2829	0.3091	2.8969	23.5577	124.7696s	393.8384
	Gibbs	0.2205	0.3375	2.7596	25.6316	43.7115	362.9895

value of all runs. The best one of each metric is highlighted in bold font. One sample run of OSPA distances are presented in Fig. 7.6.

For scenarios 1-3, the performance of these three different MHT implementation algorithms is comparable. Although MDA-MHT has lowest miscorrelation rate of true tracks, it is evident that Tabu-MHT provides the best overall performances in terms of OSPA distance and has longer true track life, which is more desirable for long-time consecutive tracking. As a comparison, Gibbs-MHT is the most efficient one among these three approaches and shows better performance in ID switch as well as correct correlation ratio. The peak values in Fig. 7.6 (a) are caused by track initialisation of new birth targets. As for the latter three random moving cases, although MDA-MHT still has lowest miscorrelation rate of true tracks, its OSPA distance is almost twice than the proposed two algorithms, leading to the fact that the proposed two MHT implementation algorithms could have more robust estimation performance. Gibbs-MHT shows best performance in estimation accuracy and is also the most efficient one. Notably, for the cases with a large number of targets (scenario 6 for example), the execution time of Gibbs-MHT is only 43.7115s, whereas Tabu-MHT takes 124.7696s and MDA-MHT requires 1403.6657s.

Based on these results, it can be concluded that the proposed Gibbs-MHT and

Tabu-MHT algorithms yield better overall performance, compared with the MDA-MHT algorithm. Note that it is clear that the number of the track hypotheses exponentially increases as the size and complexity of the problem grow. Therefore, the proposed algorithms are expected to provide better performance as the problem becomes more complex. The simulation results obtained confirm this expectation: as the complexity and the size of the scenario increases, the performance improvement becomes more significant. Furthermore, it is obvious from the simulation results that the proposed Gibbs-MHT drastically reduces the computational complexity without sacrificing the tracking performance.

7.7 Conclusions

This chapter develops two new algorithms for practical implementation of MDA with the MWIS formulation: Tabu search and Gibbs sampling implementation algorithms. The main purpose of these developments is to improve the computational efficiency while maintaining/enhancing the association performance. For the Tabu search based algorithm, random initial solutions are first generated to guarantee the diversity of the final solution. Then, two basic operators are used in each local search iteration to find better neighbourhood solutions. In the second proposed MDA implementation algorithm, the MWIS problem is first reformulated to a MPD problem to successfully apply Gibbs sampling. Then, the Gibbs sampling based algorithm is designed to find the best global hypothesis. Theoretical analysis shows that the proposed Markov chain generated by Gibbs sampler converges to its invariant distribution with an exponential rate. Simulation results of MHT applications confirm that the proposed implementation algorithms are highly efficient and provide better overall performance than the mainstream solution in most scenarios. Especially, as can be deduced, it is clear from the simulation results that the performance improvement of the proposed algorithms becomes significant in large-scale problems.

Chapter 8

Conclusions and Future Works

This thesis investigates the problem of distributed target tracking over a low-cost sensor network and proposes effective solutions for several challenging scenarios: (1) well-balance between communication cost and tracking performance in distributed STT; (2) reducing time complexity with guaranteed performance for both local MTT and distributed MTT fusion; (3) accommodating unknown environmental parameters for MTT; (4) improving the performance of MTT with closely-moving targets; and (5) fully distributed MTT with guaranteed global performance. We will first summarise the contributions of this thesis and then provide directions in which this work can be extended.

8.1 Conclusions

We propose a new SGG-DKF algorithm for distributed STT over a low-cost sensor network. The proposed algorithm utilises the concept of information weighted fusion in conjunction with a new SGG network-wide averaging process. Rigorous convergence analysis of the proposed SGG algorithm is carried out to support its applications. Theoretical analysis reveals that the proposed DKF guarantees asymptotic convergence to the optimal CKF. The prominent feature of the proposed algorithm lies in that it shows faster convergence speed, compared to the RG-DKF, and requires less communication burden, compared to the GG-DKF. The promising feature of the proposed SGG-DKF is that it offers great flexibility and well balance between communication cost and tracking accuracy.

The implementation issues for practical application of the JPDA filter are addressed

by: (1) reducing the time complexity of finding the marginal association probability; and (2) alleviating the assumption on the full knowledge of detection probability and clutter rate. By considering each possible joint hypothesis is considered as a random variable, an polynomial-time and performance-guaranteed approximation of marginalisation is proposed using Gibbs sampling technique. Incorporating Gibbs-JPDA with multi-Bernoulli filter, a new MB-JPDA for MTT with unknown detection probability and clutter rate is developed. This new algorithm is demonstrated to provide comparable performance to the ideal JPDA with full knowledge on the environmental model.

A new information-theoretic JPDA filter for tracking unknown number of closely-spaced targets in a cluttered environment. The proposed JPDA filter is derived through the minimisation of a weighted reverse KLD to approximate the posterior Gaussian mixture PDF. The resultant approximation approach turns out to show similar structure as the generalised CI in sensor fusion. Theoretical analysis of mean performance and error covariance performance with ideal detection probability is presented to provide insights of the proposed approach. Extensive empirical simulations are undertaken to validate the performance of the proposed multi-target tracking algorithm.

We propose a new fully distributed estimation algorithm for MTT over a low-cost sensor network. The algorithm developed is based on the information form JPDA filter in conjunction with a hybrid fusion strategy via the average consensus process. A distributed node counting algorithm is also proposed to support the implementation of proposed distributed MTT. The benefits of this new filter are its asymptotic convergence to the optimal centralised JPDA and local consistency preservation.

We develop new algorithms for practical implementation of MDA with MWIS formulation. Both heuristic Tabu search and stochastic Gibbs sampling are exploited to provide polynomial-time approximate MDA solution while maintaining/enhancing the association performance. For the Tabu search based algorithm, random initial solutions are generated to guarantee the diversity of the final solution. To improve the quality of each initial solution, two basic operators are suggested for the local search iteration. By transforming the MWIS problem into an equivalent MPD problem, the stochastic Gibbs sampling is also employed to efficiently find the best global hypothesis. The advantage of this algorithm is that it guarantees asymptotic convergence to the invariant target distribution with an exponential rate.

8.2 Future Works

There are several directions in which the work presented in this thesis can be extended. For the proposed SGG-DKF algorithm, it can be concluded that the sensor activation probability plays an important role in guaranteeing the convergence performance of the gossip process. Although we have theoretically analysed the effect of the sensor activation probability on the convergence rate in an average sense manner. How to choose and optimise this important parameter for a specific sensor network topology still needs further research.

The proposed DJPDA utilises the information form of the standard JPDA filter. It is known that the performance of the standard JPDA degrades drastically when targets moving closely. The information-theoretic JPDA filter developed in this thesis is demonstrated to have capability to alleviate this issue. Therefore, deriving the equivalent information form of the information-theoretic JPDA and implementing it in a distributed manner by either average consensus or gossip process is an interesting topic.

Both SGG-DKF and DJPDA are developed under ideal communication condition without any delay and link failure. Unfortunately, low-cost sensor networks might experience some undesirable communication disturbances, e.g., communication noises and communication loss/failures, induced by multi-path fading, signal attenuation, background noise, external block, etc. These unexpected factors will obviously degrade the fusion performance among the sensor nodes. How to accommodate these undesired conditions in the proposed algorithms still require further exploration.

Finally, validating the performance of our proposed techniques with real experimental data is an important issue, and needs to be addressed in the future.

References

- [1] Robert J Fitzgerald. Development of practical pda logic for multitarget tracking by microprocessor. In *1986 American Control Conference*, pages 889–898. IEEE, 1986.
- [2] Pan Quan, Zhang Hongcai, Wang Peide, and Zhou Hongren. Development of new practical multitarget data association algorithm. In *[Proceedings 1992] The First IEEE Conference on Control Applications*, pages 1065–1066. IEEE, 1992.
- [3] Babak Bakhtiar, Hossein Alavi, and Farid Amoozegar. Efficient algorithm for computing data association probabilities for multitarget tracking. In *Automatic Object Recognition VI*, volume 2756, pages 130–141. International Society for Optics and Photonics, 1996.
- [4] Bruno Silva, Roy M Fisher, Anuj Kumar, and Gerhard P Hancke. Experimental link quality characterization of wireless sensor networks for underground monitoring. *IEEE Transactions on Industrial Informatics*, 11(5):1099–1110, 2015.
- [5] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97. ACM, 2002.
- [6] Tao Sun and Ming Xin. Multiple uav target tracking using consensus-based distributed high degree cubature information filter. In *AIAA Guidance, Navigation, and Control Conference*, 2015.
- [7] Bin Jia, Khanh D Pham, Erik Blasch, Dan Shen, Zhonghai Wang, and Genshe Chen. Cooperative space object tracking using space-based optical sensors via consensus-based filters. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1908–1936, 2016.
- [8] Thanh T Vu and Amir R Rahmani. Distributed consensus-based kalman filter estimation and control of formation flying spacecraft: Simulation and validation. In *AIAA Guidance, Navigation, and Control Conference*, 2015.
- [9] Jing Yi. Sensor fusion and vehicular communication systems part ii, 2016.

- [10] Martin E Liggins, Chee-Yee Chong, Ivan Kadar, Mark G Alford, Vincent Vannicola, and Stelios Thomopoulos. Distributed fusion architectures and algorithms for target tracking. *Proceedings of the IEEE*, 85(1):95–107, 1997.
- [11] Deok-Jin Lee. Unscented information filtering for distributed estimation and multiple sensor fusion. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008.
- [12] Sailes K Sengijpta. *Fundamentals of statistical signal processing: Estimation theory*. Taylor & Francis Group, 1995.
- [13] Murtaza Taj and Andrea Cavallaro. Distributed and decentralized multicamera tracking. *IEEE Signal Processing Magazine*, 28(3):46–58, 2011.
- [14] Henry Medeiros, Johnny Park, and Avinash Kak. Distributed object tracking using a cluster-based kalman filter in wireless camera networks. *IEEE Journal of Selected Topics in Signal Processing*, 2(4):448–463, 2008.
- [15] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.
- [16] Wei Ren, Randal W Beard, and Ella M Atkins. A survey of consensus problems in multi-agent coordination. In *American Control Conference, 2005. Proceedings of the 2005*, pages 1859–1864. IEEE, 2005.
- [17] Wei Ren, Randal W Beard, and Ella M Atkins. Information consensus in multivehicle cooperative control. *IEEE Control Systems*, 27(2):71–82, 2007.
- [18] Reza Olfati-Saber and Jeff S Shamma. Consensus filters for sensor networks and distributed sensor fusion. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 6698–6703. IEEE, 2005.
- [19] Reza Olfati-Saber. Distributed kalman filtering for sensor networks. In *2007 46th IEEE Conference on Decision and Control*, pages 5492–5498. IEEE, 2007.
- [20] Reza Olfati-Saber. Kalman-consensus filter: Optimality, stability, and performance. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 7036–7042. IEEE, 2009.
- [21] Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004.
- [22] Lin Xiao, Stephen Boyd, and Sanjay Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pages 63–70. IEEE, 2005.

- [23] Ahmed Tashrif Kamal, Chong Ding, Bi Song, Jay A Farrell, and Amit K Roy-Chowdhury. A generalized kalman consensus filter for wide-area video networks. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 7863–7869. IEEE, 2011.
- [24] Reza Olfati-Saber. Distributed kalman filter with embedded consensus filters. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 8179–8184. IEEE, 2005.
- [25] Ondrej Hlinka, Ondrej Sluciak, Franz Hlawatsch, and Markus Rupp. Distributed data fusion using iterative covariance intersection. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference On*, pages 1861–1865. IEEE, 2014.
- [26] Guoliang Wei, Wangyan Li, Derui Ding, and Yurong Liu. Stability analysis of covariance intersection-based kalman consensus filtering for time-varying systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.
- [27] Wolfgang Niehsen. Information fusion based on fast covariance intersection filtering. In *Information Fusion, 2002. Proceedings of the 5th International Conference on*, pages 901–904. IEEE, 2002.
- [28] Giorgio Battistelli and Luigi Chisci. Kullback–leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability. *Automatica*, 50(3):707–718, 2014.
- [29] Joris Sijs and Mircea Lazar. State fusion with unknown correlation: Ellipsoidal intersection. *Automatica*, 8(48):1874–1878, 2012.
- [30] Benjamin Noack, Joris Sijs, Marc Reinhardt, and Uwe D Hanebeck. Decentralized data fusion with inverse covariance intersection. *Automatica*, 79:35–41, 2017.
- [31] Ahmed Tashrif Kamal, Jay A Farrell, and Amit K Roy-Chowdhury. Information weighted consensus. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 2732–2737. IEEE, 2012.
- [32] Ahmed T Kamal, Jay A Farrell, and Amit K Roy-Chowdhury. Information weighted consensus filters and their application in distributed camera networks. *IEEE Transactions on Automatic Control*, 58(12):3112–3125, 2013.
- [33] Peng Yao, Gang Liu, and Yanfei Liu. Average information-weighted consensus filter for target tracking in distributed sensor networks with naivety issues. *International Journal of Adaptive Control and Signal Processing*, 32(5):681–699, 2018.
- [34] Giorgio Battistelli, Luigi Chisci, Giovanni Mugnai, Alfonso Farina, and Antonio Graziano. Consensus-based linear and nonlinear filtering. *IEEE Transactions on Automatic Control*, 60(5):1410–1415, 2015.

- [35] Kangjian Ma, Shaochuan Wu, Yuming Wei, and Wenbin Zhang. Gossip-based distributed tracking in networks of heterogeneous agents. *IEEE Communications Letters*, 21(4):801–804, 2017.
- [36] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530, 2006.
- [37] Soummya Kar and José MF Moura. Gossip and distributed kalman filtering: Weak consensus under weak detectability. *IEEE Transactions on Signal Processing*, 59(4):1766–1784, 2011.
- [38] Di Li, Soummya Kar, José MF Moura, H Vincent Poor, and Shuguang Cui. Distributed kalman filtering over massive data sets: analysis through large deviations of random riccati equations. *IEEE Transactions on Information Theory*, 61(3):1351–1372, 2015.
- [39] Jiahu Qin, Jie Wang, Ling Shi, and Yu Kang. Randomized consensus based distributed kalman filtering over wireless sensor networks. *arXiv preprint arXiv:1810.02531*, 2018.
- [40] Chao Wan, Yongxin Gao, X Rong Li, and Enbin Song. Distributed filtering over networks using greedy gossip. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1968–1975. IEEE, 2018.
- [41] Federico S Cattivelli, Cassio G Lopes, and Ali H Sayed. A diffusion rls scheme for distributed estimation over adaptive networks. In *Signal Processing Advances in Wireless Communications, 2007. SPAWC 2007. IEEE 8th Workshop on*, pages 1–5. IEEE, 2007.
- [42] Federico S Cattivelli and Ali H Sayed. Diffusion strategies for distributed kalman filtering and smoothing. *IEEE Transactions on Automatic Control*, 55(9):2069–2084, 2010.
- [43] Jinwen Hu, Lihua Xie, and Cishen Zhang. Diffusion kalman filtering based on covariance intersection. *IEEE Transactions on Signal Processing*, 60(2):891–902, 2012.
- [44] Yonggang Zhang, Chengcheng Wang, Ning Li, and Jonathon Chambers. Diffusion kalman filter based on local estimate exchanges. In *Digital Signal Processing (DSP), 2015 IEEE International Conference on*, pages 828–832. IEEE, 2015.
- [45] Shaocheng Wang and Wei Ren. On the convergence conditions of distributed dynamic state estimation using sensor networks: A unified framework. *IEEE Transactions on Control Systems Technology*, 26(4):1300–1316, 2018.
- [46] Ruggero Carli, Alessandro Chiuso, Luca Schenato, and Sandro Zampieri. Distributed kalman filtering based on consensus strategies. *IEEE Journal on Selected Areas in Communications*, 26(4):622–633, 2008.

- [47] Maryam Kamgarpour and Claire Tomlin. Convergence properties of a decentralized kalman filter. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 3205–3210. IEEE, 2008.
- [48] Wangyan Li, Guoliang Wei, Daniel WC Ho, and Derui Ding. A weightedly uniform detectability for sensor networks. *IEEE Transactions on Neural Networks and Learning Systems*, 29(11):5790–5796, 2018.
- [49] R Singer, R Sea, and K Housewright. Derivation and evaluation of improved tracking filter for use in dense multitarget environments. *IEEE Transactions on Information Theory*, 20(4):423–432, 1974.
- [50] Samuel S Blackman. Multiple-target tracking with radar applications. *Dedham, MA, Artech House, Inc., 1986, 463 p.*, 1986.
- [51] Thomas Fortmann, Yaakov Bar-Shalom, and Molly Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8(3):173–184, 1983.
- [52] Yaakov Bar-Shalom, Fred Daum, and Jim Huang. The probabilistic data association filter. *IEEE Control Systems Magazine*, 29(6):82–100, 2009.
- [53] Darko Musicki, Robin Evans, and Srdjan Stankovic. Integrated probabilistic data association. *IEEE Transactions on Automatic Control*, 39(6):1237–1241, 1994.
- [54] Subhash Challa, Mark R Morelande, Darko Mušicki, and Robin J Evans. *Fundamentals of object tracking*. Cambridge University Press, 2011.
- [55] Darko Musicki and Robin Evans. Joint integrated probabilistic data association: Jipda. *IEEE Transactions on Aerospace and Electronic Systems*, 40(3):1093–1099, 2004.
- [56] Seyed Hamid Rezatofighi, Anton Milan, Zhen Zhang, Qinfeng Shi, Anthony Dick, and Ian Reid. Joint probabilistic data association revisited. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3047–3055. IEEE, 2015.
- [57] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.
- [58] Lennart Svensson, Daniel Svensson, Marco Guerriero, and Peter Willett. Set jpda filter for multitarget tracking. *IEEE Transactions on Signal Processing*, 59(10):4677–4691, 2011.
- [59] Jason L Williams. Marginal multi-bernoulli filters: Rfs derivation of mht, jipda, and association-based member. *IEEE Transactions on Aerospace and Electronic Systems*, 51(3):1664–1687, 2015.

- [60] Roslyn A Lau and Jason L Williams. A structured mean field approach for existence-based multiple target tracking. In *Information Fusion (FUSION), 2016 19th International Conference on*, pages 1111–1118. IEEE, 2016.
- [61] Donald Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979.
- [62] Martin R Stiglitiz and Christine Blanchard. Multitarget-multisensor tracking: Advanced applications. *Microwave Journal*, 33(8):160–161, 1990.
- [63] Ingemar J. Cox and Sunita L. Hingorani. An efficient implementation of reid’s multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):138–150, 1996.
- [64] Katta G Murty. An algorithm for ranking all the assignments in order of increasing cost. *Operations research*, 16(3):682–687, 1968.
- [65] Somnath Deb, Krishna R Pattipati, and Yaakov Bar-Shalom. A multisensor-multitarget data association algorithm for heterogeneous sensors. *IEEE Transactions on Aerospace and Electronic Systems*, 29(2):560–568, 1993.
- [66] Aubrey B Poore and Alexander J Robertson III. A new lagrangian relaxation based algorithm for a class of multidimensional assignment problems. *Computational Optimization and Applications*, 8(2):129–150, 1997.
- [67] Robert T Collins. Multitarget data association with higher-order motion models. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1744–1751. IEEE, 2012.
- [68] Samuel Blackman and Robert Popoli. Design and analysis of modern tracking systems (artech house radar library). *Artech house*, 1999.
- [69] Javier Areta, Yaakov Bar-Shalom, Mark Levedahl, and Krishna R Pattipati. Hierarchical track association and fusion for a networked surveillance system. *Journal of Advances in Information Fusion*, 1(2):140–157, 2006.
- [70] Dimitri J Papageorgiou and Michael R Salpukas. The maximum weight independent set problem for data association in multiple hypothesis tracking. In *Optimization and Cooperative Control Strategies*, pages 235–255. Springer, 2009.
- [71] William Brendel, Mohamed Amer, and Sinisa Todorovic. Multiobject tracking as maximum weight independent set. In *CVPR 2011*, pages 1273–1280. IEEE, 2011.
- [72] Chanh Kim, Fuxin Li, Arridhana Ciptadi, and James M Rehg. Multiple hypothesis tracking revisited. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4696–4704, 2015.

-
- [73] Patric RJ Östergård. A new algorithm for the maximum-weight clique problem. *Nordic Journal of Computing*, 8(4):424–436, 2001.
 - [74] Sk Md Abu Nayeem and Madhumangal Pal. Genetic algorithmic approach to find the maximum weight independent set of a graph. *Journal of Applied Mathematics and Computing*, 25(1-2):217–229, 2007.
 - [75] Bruno Nogueira, Rian GS Pinheiro, and Anand Subramanian. A hybrid iterated local search heuristic for the maximum weight independent set problem. *Optimization Letters*, 12(3):567–583, 2018.
 - [76] B-N Vo, Sumeetpal Singh, and Arnaud Doucet. Sequential monte carlo methods for multitarget filtering with random finite sets. *IEEE Transactions on Aerospace and Electronic Systems*, 41(4):1224–1245, 2005.
 - [77] Nick Whiteley, Sumeetpal Singh, and Simon Godsill. Auxiliary particle implementation of probability hypothesis density filter. *IEEE Transactions on Aerospace and Electronic Systems*, 46(3):1437–1454, 2010.
 - [78] Ba-Tuong Vo, Ba-Ngu Vo, and Antonio Cantoni. Analytic implementations of the cardinalized probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 55(7):3553–3567, 2007.
 - [79] Ba-Tuong Vo, Ba-Ngu Vo, and Antonio Cantoni. The cardinality balanced multi-target multi-bernoulli filter and its implementations. *IEEE Transactions on Signal Processing*, 57(2):409–423, 2009.
 - [80] Ba-Tuong Vo, Ba-Ngu Vo, Reza Hoseinnezhad, and Ronald PS Mahler. Robust multi-bernoulli filtering. *IEEE Journal of Selected Topics in Signal Processing*, 7(3):399–409, 2013.
 - [81] Ba-Ngu Vo, Ba-Tuong Vo, and Dinh Phung. Labeled random finite sets and the bayes multi-target tracking filter. *IEEE Transactions on Signal Processing*, 62(24):6554–6567, 2014.
 - [82] Ba-Tuong Vo and Ba-Ngu Vo. Labeled random finite sets and multi-object conjugate priors. *IEEE Transactions on Signal Processing*, 61(13):3460–3475, 2013.
 - [83] Ahmed Elgammal, Ramani Duraiswami, David Harwood, and Larry S Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90(7):1151–1163, 2002.
 - [84] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.
 - [85] X Rong Li and Ning Li. Integrated real-time estimation of clutter density for tracking. *IEEE Transactions on Signal Processing*, 48(10):2797–2805, 2000.

- [86] Taek Lyul Song and Darko Musicki. Adaptive clutter measurement density estimation for improved target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 47(2):1457–1466, 2011.
- [87] Xin Chen, Ratnasingham Tharmarasa, Thia Kirubarajan, and Mike McDonald. Online clutter estimation using a gaussian kernel density estimator for multitarget tracking. *IET Radar, Sonar & Navigation*, 9(1):1–9, 2014.
- [88] Woo Chan Kim and Taek Lyul Song. Spatial clutter measurement density estimation in nonhomogeneous measurement spaces. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 1772–1777. IEEE, 2015.
- [89] Ronald PS Mahler, Ba-Tuong Vo, and Ba-Ngu Vo. Cphd filtering with unknown clutter rate and detection profile. *IEEE Transactions on Signal Processing*, 59(8):3497–3513, 2011.
- [90] Samir Hachour, François Delmotte, and David Mercier. A distributed solution for multi-object tracking and classification. In *17th International Conference on Information Fusion (FUSION)*, pages 1–7. IEEE, 2014.
- [91] Yihua Yu. Distributed target tracking in wireless sensor networks with data association uncertainty. *IEEE Communications Letters*, 21(6):1281–1284, 2017.
- [92] Shaoming He, Hyo-Sang Shin, and Antonios Tsourdos. Distributed joint probabilistic data association filter with hybrid fusion strategy. *IEEE Transactions on Instrumentation and Measurement*, 2019.
- [93] Nils F Sandell and Reza Olfati-Saber. Distributed data association for multi-target tracking in sensor networks. In *2008 47th IEEE Conference on Decision and Control*, pages 1085–1090. IEEE, 2008.
- [94] Yanming Chen, Qingjie Zhao, Zhulin An, Peng Lv, and Liujun Zhao. Distributed multi-target tracking based on the k-mtsfc algorithm in camera networks. *IEEE Sensors Journal*, 16(13):5481–5490, 2016.
- [95] Ahmed T Kamal, Jawadul H Bappy, Jay A Farrell, and Amit K Roy-Chowdhury. Distributed multi-target tracking and data association in vision networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1397–1410, 2016.
- [96] Tiancheng Li, Juan M Corchado, and Shudong Sun. Partial consensus and conservative fusion of gaussian mixtures for distributed phd fusion. *IEEE Transactions on Aerospace and Electronic Systems*, 2018.
- [97] Giorgio Battistelli, Luigi Chisci, Claudio Fantacci, Alfonso Farina, and Antonio Graziano. Consensus cphd filter for distributed multitarget tracking. *IEEE Journal of Selected Topics in Signal Processing*, 7(3):508–520, 2013.
- [98] Long Liu, Hongbing Ji, and Zhenhua Fan. A cardinality modified product multi-sensor phd. *Information Fusion*, 31:87–99, 2016.

- [99] Lucy Y Pao and Lidija Trailovic. The optimal order of processing sensor information in sequential multisensor fusion algorithms. *IEEE Transactions on Automatic Control*, 45(8):1532–1536, 2000.
- [100] Lidija Trailovic and Lucy Y Pao. Computing budget allocation for efficient ranking and selection of variances with application to target tracking algorithms. *IEEE Transactions on Automatic Control*, 49(1):58–67, 2004.
- [101] Taek Song, Hyoung Kim, and Darko Musicki. Distributed (nonlinear) target tracking in clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 51(1):654–668, 2015.
- [102] Tiancheng Li, Juan M Corchado, and Javier Prieto. Convergence of distributed flooding and its application for distributed bayesian filtering. *IEEE Transactions on Signal and Information Processing over Networks*, 3(3):580–591, 2017.
- [103] Deniz Ustebay, Boris N Oreshkin, Mark J Coates, and Michael G Rabbat. Greedy gossip with eavesdropping. *IEEE Transactions on Signal Processing*, 58(7):3765–3776, 2010.
- [104] David Kempe, Alin Dobra, and Johannes Gehrke. Gossip-based computation of aggregate information. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 482–491. IEEE, 2003.
- [105] Richard Karp, Christian Schindelhauer, Scott Shenker, and Berthold Vocking. Randomized rumor spreading. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 565–574. IEEE, 2000.
- [106] Giorgio Battistelli, Luigi Chisci, Giovanni Mugnai, Alfonso Farina, and Antonio Graziano. Consensus-based linear and nonlinear filtering. *IEEE Transactions on Automatic Control*, 60(5):1410–1415, 2015.
- [107] Songhwai Oh, Stuart Russell, and Shankar Sastry. Markov chain monte carlo data association for general multiple-target tracking problems. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, pages 735–742. IEEE, 2004.
- [108] David F Crouse, Yaakov Bar-Shalom, Peter Willett, and Lennart Svensson. The jpdaf in practical systems: Computation and snake oil. In *Signal and Data Processing of Small Targets 2010*, volume 7698, page 769813. International Society for Optics and Photonics, 2010.
- [109] JA Roecker and GL Phillis. Suboptimal joint probabilistic data association. *IEEE Transactions on Aerospace and Electronic Systems*, 29(2):510–517, 1993.
- [110] James A Roecker. A class of near optimal jpda algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 30(2):504–510, 1994.

- [111] Ba-Ngu Vo, Ba-Tuong Vo, and Hung Gia Hoang. An efficient implementation of the generalized labeled multi-bernoulli filter. *IEEE Transactions on Signal Processing*, 65(8):1975–1987, 2017.
- [112] George Casella and Edward I George. Explaining the gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.
- [113] Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of markov chain monte carlo, Chapter 5*. CRC press, 2011.
- [114] Robert G Gallager. *Stochastic processes: theory for applications*. Cambridge University Press, 2013.
- [115] C Ouyang, H Ji, and C Li. Improved multi-target multi-bernoulli filter. *IET Radar, Sonar & Navigation*, 6(6):458–464, 2012.
- [116] Dominic Schuhmacher, Ba-Tuong Vo, and Ba-Ngu Vo. A consistent metric for performance evaluation of multi-object filters. *IEEE Transactions on Signal Processing*, 56(8):3447–3457, 2008.
- [117] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.
- [118] Andrew R Runnalls. Kullback-leibler approach to gaussian mixture reduction. *IEEE Transactions on Aerospace and Electronic Systems*, 43(3), 2007.
- [119] Tohid Ardeshiri, Umut Orguner, and Emre Özkan. Gaussian mixture reduction using reverse kullback-leibler divergence. *arXiv preprint arXiv:1508.05514*, 2015.
- [120] Paul Bromiley. Products and convolutions of gaussian probability density functions. *Tina-Vision Memo*, 3(4), 2003.
- [121] João Sequeira, Antonios Tsourdos, and Samuel B Lazarus. Robust covariance estimation for data fusion from multiple sensors. *IEEE Transactions on Instrumentation and Measurement*, 60(12):3833–3844, 2011.
- [122] Thomas H Cormen. *Introduction to algorithms*. MIT press, 2009.
- [123] Deepa Elizabeth George and A Unnikrishnan. On the divergence of information filter for multi sensors fusion. *Information Fusion*, 27:76–84, 2016.
- [124] Jason Stauch, Travis Bessell, Mark Rutten, Jason Baldwin, Moriba Jah, and Keric Hill. Joint probabilistic data association and smoothing applied to multiple space object tracking. *Journal of Guidance, Control, and Dynamics*, 41(1):19–33, 2018.
- [125] Darko Mušicki, Taek Lyul Song, Hae Ho Lee, Xin Chen, and Thialingam Kirubarajan. Track-to-track fusion with target existence. *IET Radar, Sonar & Navigation*, 9(3):241–248, 2014.

- [126] Giorgio Battistelli, Luigi Chisci, Francesco Papi, Alessio Benavoli, Alfonso Farina, and Antonio Graziano. Optimal flow models for multiscan data association. *IEEE Transactions on Aerospace and Electronic Systems*, 47(4):2405–2422, 2011.
- [127] Somnath Deb, Murali Yeddanapudi, Krishna Pattipati, and Yaakov Bar-Shalom. A generalized sd assignment algorithm for multisensor-multitarget state estimation. *IEEE Transactions on Aerospace and Electronic Systems*, 33(2):523–538, 1997.
- [128] Stephen Boyd, Persi Diaconis, Jun Sun, and Lin Xiao. Fastest mixing markov chain on a path. *The American Mathematical Monthly*, 113(1):70–74, 2006.
- [129] Franck Iutzeler, Philippe Ciblat, and Jérémie Jakubowicz. Analysis of max-consensus algorithms in wireless channels. *IEEE Transactions on Signal Processing*, 60(11):6103–6107, 2012.
- [130] Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1806–1819, 2011.
- [131] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.
- [132] Yuanlu Xu, Xiaobai Liu, Yang Liu, and Song-Chun Zhu. Multi-view people tracking via hierarchical trajectory composition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4256–4265, 2016.
- [133] Panos M Pardalos, L Pitsoulis, T Mavridou, and Mauricio GC Resende. Parallel search for combinatorial optimization: Genetic algorithms, simulated annealing, tabu search and grasp. In *International Workshop on Parallel Algorithms for Irregularly Structured Problems*, pages 317–331. Springer, 1995.
- [134] Roberto Battiti and Giampietro Tecchiolli. The continuous reactive tabu search: blending combinatorial optimization and stochastic search for global optimization. *Annals of Operations Research*, 63(2):151–188, 1996.
- [135] Emile Aarts, Emile HL Aarts, and Jan Karel Lenstra. *Local search in combinatorial optimization*. Princeton University Press, 2003.
- [136] Yaakov Bar-Shalom, Sam S Blackman, and Robert J Fitzgerald. Dimensionless score function for multiple hypothesis tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 43(1):392–400, 2007.
- [137] Lance M Kaplan, Yaakov Bar-Shalom, and William D Blair. Assignment costs for multiple sensor track-to-track association. *IEEE Transactions on Aerospace and Electronic Systems*, 44(2):655–677, 2008.

-
- [138] Yaakov Bar-Shalom and Huimin Chen. Multisensor track-to-track association for tracks with dependent errors. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 3, pages 2674–2679. IEEE, 2004.
 - [139] Fred Glover. Tabu search: A tutorial. *Interfaces*, 20(4):74–94, 1990.
 - [140] Henry Leung, Zhijian Hu, and Martin Blanchette. Evaluation of multiple radar target trackers in stressful environments. *IEEE Transactions on Aerospace and Electronic Systems*, 35(2):663–674, 1999.
 - [141] J Deyst and C Price. Conditions for asymptotic stability of the discrete minimum-variance linear estimator. *IEEE Transactions on Automatic Control*, 13(6):702–705, 1968.

Appendix A

Pseudo Codes of RG-DKF and GG-DKF

Algorithm A.1 Randomised gossip distributed Kalman filter at the i th node

Input: Previous target estimation $\{\mathbf{x}_{k-1|k-1,i}, \mathbf{P}_{k-1|k-1,i}\}$, received measurements $\mathbf{z}_{k,i}$

Output: Current estimation $\{\mathbf{x}_{k|k,i}, \mathbf{P}_{k|k,i}\}$

(1) Prediction:

$$\begin{aligned}\mathbf{x}_{k|k-1,i} &= \mathbf{F}_k \mathbf{x}_{k-1|k-1,i} \\ \mathbf{P}_{k|k-1,i} &= \mathbf{F}_{k-1} \mathbf{P}_{k-1|k-1,i} \mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}\end{aligned}$$

(2) Compute the information terms:

$$\begin{aligned}\mathbf{u}_{k,i} &= \frac{1}{N} \mathbf{P}_{k|k-1,i}^{-1} \mathbf{x}_{k|k-1,i} + \mathbf{H}_{k,i}^T \mathbf{R}_{k,i}^{-1} \mathbf{z}_{k,i} \\ \mathbf{U}_{k,i} &= \frac{1}{N} \mathbf{P}_{k|k-1,i}^{-1} + \mathbf{H}_{k,i}^T \mathbf{R}_{k,i}^{-1} \mathbf{H}_{k,i}\end{aligned}$$

(3) **for** $l = 0, 1, \dots, L$ **do**

 Randomly pick up a neighbour $j \in \mathcal{N}_i$

 Perform information averaging

$$\begin{aligned}\mathbf{u}_{k,i}(l) &= \frac{1}{2} [\mathbf{u}_{k,i}(l-1) + \mathbf{u}_{k,j^*}(l-1)] \\ \mathbf{U}_{k,i}(l) &= \frac{1}{2} [\mathbf{U}_{k,i}(l-1) + \mathbf{U}_{k,j^*}(l-1)]\end{aligned}$$

(4) Measurement update:

$$\begin{aligned}\mathbf{x}_{k|k,i} &= [\mathbf{U}_{k,i}(L)]^{-1} \mathbf{u}_{k,i}(L) \\ \mathbf{P}_{k|k,i}^{-1} &= N \mathbf{U}_{k,i}(L)\end{aligned}$$

Algorithm A.2 Greedy gossip distributed Kalman filter at the i th node

Input: Previous target estimation $\{\mathbf{x}_{k-1|k-1,i}, \mathbf{P}_{k-1|k-1,i}\}$, received measurements $\mathbf{z}_{k,i}$

Output: Current estimation $\{\mathbf{x}_{k|k,i}, \mathbf{P}_{k|k,i}\}$

(1) Prediction:

$$\begin{aligned}\mathbf{x}_{k|k-1,i} &= \mathbf{F}_k \mathbf{x}_{k-1|k-1,i} \\ \mathbf{P}_{k|k-1,i} &= \mathbf{F}_{k-1} \mathbf{P}_{k-1|k-1,i} \mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}\end{aligned}$$

(2) Compute the information terms:

$$\begin{aligned}\mathbf{u}_{k,i} &= \frac{1}{N} \mathbf{P}_{k|k-1,i}^{-1} \mathbf{x}_{k|k-1,i} + \mathbf{H}_{k,i}^T \mathbf{R}_{k,i}^{-1} \mathbf{z}_{k,i} \\ \mathbf{U}_{k,i} &= \frac{1}{N} \mathbf{P}_{k|k-1,i}^{-1} + \mathbf{H}_{k,i}^T \mathbf{R}_{k,i}^{-1} \mathbf{H}_{k,i}\end{aligned}$$

(3) **for** $l = 0, 1, \dots, L$ **do**

Compute the Mahalanobis distance $d_{i,j}$

Find a local neighbour j^* which has the largest value of $d_{i,j^*}(l)$ among the active sensor nodes as

$$j^* = \max_j d_{i,j}(l), \quad j \in \mathcal{N}_i$$

After finding j^* , perform information averaging

$$\begin{aligned}\mathbf{u}_{k,i}(l) &= \frac{1}{2} [\mathbf{u}_{k,i}(l-1) + \mathbf{u}_{k,j^*}(l-1)] \\ \mathbf{U}_{k,i}(l) &= \frac{1}{2} [\mathbf{U}_{k,i}(l-1) + \mathbf{U}_{k,j^*}(l-1)]\end{aligned}$$

(4) Measurement update:

$$\begin{aligned}\mathbf{x}_{k|k,i} &= [\mathbf{U}_{k,i}(L)]^{-1} \mathbf{u}_{k,i}(L) \\ \mathbf{P}_{k|k,i}^{-1} &= N \mathbf{U}_{k,i}(L)\end{aligned}$$

Appendix B

Properties of Finite-State Markov Chain

This appendix collects two key lemmas in [114] that are used in the proof of Theorem 4.1.

Lemma B.1. *Let π be the transition operator of a finite-state Markov chain and let π^n be the n th order transition probabilities. Then, for any state and $n \geq 1$, we have*

$$\begin{aligned}\max_{\boldsymbol{\theta}} \pi^{n+1}(\boldsymbol{\chi}|\boldsymbol{\theta}) &\leq \max_{\boldsymbol{\varsigma}} \pi^n(\boldsymbol{\chi}|\boldsymbol{\varsigma}) \\ \min_{\boldsymbol{\theta}} \pi^{n+1}(\boldsymbol{\chi}|\boldsymbol{\theta}) &\geq \min_{\boldsymbol{\varsigma}} \pi^n(\boldsymbol{\chi}|\boldsymbol{\varsigma})\end{aligned}\tag{B.1}$$

Lemma B.2. *Let $\alpha = \min_{\boldsymbol{\theta}} \pi(\boldsymbol{\chi}|\boldsymbol{\theta})$ and the transition probability of a finite-state Markov chain for any given state to another state is larger than zero. Then, for any state and $n \geq 1$, we have*

$$\begin{aligned}\max_{\boldsymbol{\theta}} \pi^{n+1}(\boldsymbol{\chi}|\boldsymbol{\theta}) - \min_{\boldsymbol{\theta}} \pi^{n+1}(\boldsymbol{\chi}|\boldsymbol{\theta}) &\leq \left[\max_{\boldsymbol{\varsigma}} \pi^n(\boldsymbol{\chi}|\boldsymbol{\varsigma}) - \min_{\boldsymbol{\varsigma}} \pi^n(\boldsymbol{\chi}|\boldsymbol{\varsigma}) \right] (1 - 2\alpha) \\ \max_{\boldsymbol{\varsigma}} \pi^n(\boldsymbol{\chi}|\boldsymbol{\varsigma}) - \min_{\boldsymbol{\varsigma}} \pi^n(\boldsymbol{\chi}|\boldsymbol{\varsigma}) &\leq (1 - 2\alpha)^n \\ \lim_{n \rightarrow \infty} \max_{\boldsymbol{\varsigma}} \pi^n(\boldsymbol{\chi}|\boldsymbol{\varsigma}) &= \lim_{n \rightarrow \infty} \min_{\boldsymbol{\varsigma}} \pi^n(\boldsymbol{\chi}|\boldsymbol{\varsigma}) \geq \alpha > 0\end{aligned}\tag{B.2}$$

Appendix C

Boundedness of Kalman Filter

This appendix collects a useful lemma regarding the boundedness of Kalman filter from [141]. Consider the following linear system

$$\begin{aligned}\mathbf{x}_k &= \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \\ \mathbf{z}_k &= \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k\end{aligned}\tag{C.1}$$

where $\mathbf{x}_k \in \mathbb{R}^n$ and $\mathbf{z}_k \in \mathbb{R}^m$ denote the system state and the corresponding measurement at time step k , respectively. The notations $\mathbf{F}_{k-1} \in \mathbb{R}^{n \times n}$ and $\mathbf{H}_k \in \mathbb{R}^{m \times n}$ correspond to the system matrix, input matrix and observation matrix, respectively. The signals $\mathbf{w}_k \in \mathbb{R}^n$ and $\mathbf{v}_k \in \mathbb{R}^m$ are process noise and measurement noise, which are assumed to be zero-mean Gaussian process with covariances $\mathbf{Q}_k \in \mathbb{R}^{n \times n}$ and $\mathbf{R}_k \in \mathbb{R}^{m \times m}$, respectively.

Definition C.1. *System (C.1) is uniformly controllable if there exist a positive integer k_0 and positive constants $\kappa_1 \leq \kappa_2 < \infty$ such that*

$$\kappa_1 \mathbf{I}_n \leq \sum_{i=k-k_0+1}^k \Phi_{k,i} \mathbf{Q}_i \Phi_{k,i}^T \leq \kappa_2 \mathbf{I}_n \tag{C.2}$$

where

$$\Phi_{k,l} = \begin{cases} \mathbf{F}_{k-1} \mathbf{F}_{k-2} \cdots \mathbf{F}_l, & k > l \\ \mathbf{I}_n, & k = l \end{cases} \tag{C.3}$$

Definition C.2. *System (C.1) is uniformly observable if there exist a positive integer k_0 and positive constants $\kappa_3 \leq \kappa_4 < \infty$ such that*

$$\kappa_3 \mathbf{I}_m \leq \sum_{i=k-k_0}^k \Phi_{i,k}^T \mathbf{H}_i^T \mathbf{R}_i^{-1} \mathbf{H}_i \Phi_{i,k} \leq \kappa_4 \mathbf{I}_m \tag{C.4}$$

where

$$\Phi_{i,k} = \Phi_{k,i}^{-1} \tag{C.5}$$

By applying Kalman filter to system (C.1), we can have the following result.

Lemma C.1. *Suppose system (C.1) is uniformly controllable and uniformly observable. If the initial error covariance $\mathbf{P}_{0|0}$ is positive, then there exists a positive integer k_0 such that the error covariance provided by Kalman filter is uniformly bounded from below and from above for all $k \geq k_0$ as*

$$\underline{\mathbf{P}} \leq \mathbf{P}_{k|k} \leq \overline{\mathbf{P}} \quad (\text{C.6})$$

where $\underline{\mathbf{P}}$ and $\overline{\mathbf{P}}$ are positive matrices.

As a consequence, the estimation error given by Kalman filter is asymptotically bounded in a mean square sense in that

$$\lim_{k \rightarrow \infty} \mathbb{E} \left[(\mathbf{x}_k - \mathbf{x}_{k|k})^T (\mathbf{x}_k - \mathbf{x}_{k|k}) \right] \leq \text{trace}(\overline{\mathbf{P}}) \quad (\text{C.7})$$

Appendix D

Some Approximated Marginalisation Algorithms

This appendix collects some approximated marginalisation algorithms that are used in Chapter 4 for comparison.

D.1 ENNJPDA [1]

For a given track, ENNJPDA finds one measurement which provides the highest marginal association probability. The selected track and its corresponding matched measurement are then removed from consideration and the measurement selection procedure is repeated for the next track until all tracks have their matched measurements. In other words, ENNJPDA only use one specific measurement in updating one track. During the measurement selection procedure, ENNJPDA utilises the concept of cheap JPDA in calculating the marginal association probability. The cheap JPDA leverages an *ad hoc* formula as

$$\beta_j^i = \frac{G_j^i}{\sum_{i=1}^{N_k} G_j^i + \sum_{j=0}^{M_k} G_j^i - G_j^i + B} \quad (\text{D.1})$$

with G_j^i is a likelihood function, which qualifies how well the j th measurement matches the i th target and is defined as

$$G_j^i = \frac{1}{\sqrt{\det(\mathbf{S}_k^i)}} \exp \left(-\frac{1}{2} (\mathbf{z}_k^j - \hat{\mathbf{z}}_k^i)^T (\mathbf{S}_k^i)^{-1} (\mathbf{z}_k^j - \hat{\mathbf{z}}_k^i) \right) \quad (\text{D.2})$$

where $\hat{\mathbf{z}}_k^i$ denotes the measurement prediction of target i at scan k and $\mathbf{S}_k^i$ stands for the covariance of innovation. The parameter $B \in (0, 1)$ is a bias term that is empirically set.

D.2 Modified Cheap JPDA [2]

The modified cheap JPDA includes an amendment coefficient r_j^i in the calculation of the marginal association probability as

$$\beta_j^i = \frac{G_j^i}{\sum_{i=1}^{N_k} G_j^i + \sum_{j=0}^{M_k} r_j^i G_j^i - r_j^i G_j^i + B} \quad (\text{D.3})$$

where the coefficient r_j^i is given by

$$r_j^i = \frac{G_j^i}{\sum_{j=0}^{M_k} r_j^i G_j^i} \quad (\text{D.4})$$

D.3 Suboptimal JPDA [3]

Suboptimal JPDA decouples the marginal association probability as

$$\beta_j^i \propto G_j^i F_j^i \quad (\text{D.5})$$

where F_j^i is a function of all $G_{j'}^{i'}$ for $j' \neq j$ and $i' \neq i$.

Define a likelihood matrix

$$\mathbf{P} = \begin{bmatrix} G_{0,1} & G_{0,2} & \cdots & G_{0,N_k} \\ G_{1,1} & G_{1,2} & \cdots & G_{1,N_k} \\ \vdots & \vdots & & \vdots \\ G_{M_k,1} & G_{M_k,2} & \cdots & G_{M_k,N_k} \end{bmatrix} \quad (\text{D.6})$$

The value of F_j^i is then given by the product of sum of each column, omitting row j and column i . For example, F_1^1 can be calculated as

$$F_{11} = (G_{0,2} + G_{2,2} + G_{3,2} + \cdots + G_{M_k,2}) (G_{0,3} + G_{2,3} + \cdots + G_{M_k,3}) \cdots \\ \times (G_{0,N_k} + G_{2,N_k} + G_{3,N_k} + \cdots + G_{M_k,N_k}) \quad (\text{D.7})$$

Then, the marginal association probability can be approximated as

$$\beta_j^i = \frac{1}{c} G_j^i \prod_{i'=1; i' \neq i}^{N_k} \sum_{j'=0; j' \neq j}^{M_k} G_{j'}^{i'} \quad (\text{D.8})$$

where c is a normalisation constant.