

TRACKING IN DISTRIBUTED NETWORKS  
USING HARMONIC MEAN DENSITY

TRACKING IN DISTRIBUTED NETWORKS USING HARMONIC  
MEAN DENSITY

BY  
NIKHIL SHARMA, M.Tech.

A THESIS  
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING  
AND THE SCHOOL OF GRADUATE STUDIES  
OF MCMASTER UNIVERSITY  
IN PARTIAL FULFILMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY

© Copyright by Nikhil Sharma, September 2023

All Rights Reserved

Doctor of Philosophy (2023)  
(Electrical & Computer Engineering)

McMaster University  
Hamilton, Ontario, Canada

TITLE: Tracking in Distributed Networks using Harmonic Mean  
Density

AUTHOR: Nikhil Sharma  
M. Tech. (Mechatronics Engineering),  
Indian Institute of Technology Patna, India

SUPERVISOR: Dr. T. Kirubarajan

NUMBER OF PAGES: xvi, 246

# Abstract

Sensors are getting smaller, inexpensive and sophisticated, with an increased availability. Compared to 25 years ago, an object tracking system now can easily achieve twice the accuracy, a much larger coverage and fault tolerance, without any significant changes in the overall cost. This is possible by simply employing more than just one sensor and processing measurements from individual sensors sequentially (or even in a batch form).

In sophisticated sensors, the number of detections can reach thousands in a single frame. The communication and computation load for gathering all such detections at the fusion center will hamper the system's performance while also being vulnerable to faults. A better solution is a distributed architecture wherein the individual sensors are equipped with processing capabilities such that they can detect measurements, extract clutter, form tracks and transmit them to the fusion center. The fusion center now fuses tracks instead of measurements, due to which this scheme is commonly termed track-level fusion.

In addition to sub-optimality, the track-level fusion suffers from a very coarse problem, which occurs due to correlations between the tracks to be fused. Often, in realistic scenarios, the cross-correlations are unknown, without any means to calculate them. Thus, fusion cannot be performed using traditional methods unless extra

information is transmitted from the fusion center.

This thesis proposes a novel and generalized method of fusing any two probability density functions (pdf) such that a positive cross-correlation exists between them. In modern tracking systems, the tracks are essentially pdfs and not necessarily Gaussian. We propose harmonic mean density based fusion and prove that it obeys all the necessary requirements of being a viable fusion mechanism. We show that fusion in this case is a classical example of agreement between the fused and participating densities based on average  $\chi^2$  divergence. Compared to other such fusion techniques in the literature, the HMD performs exceptionally well.

Transmitting covariance matrices in distributed architecture is not always possible in cases for e.g. tactical and automotive systems. Fusion of tracks without the knowledge of uncertainty is another problem discussed in the thesis. We propose a novel technique for local covariance reconstruction at the fusion center with the knowledge of estimates and a vector of times when update has occurred at local sensor node. It has been shown on a realistic scenario that the reconstructed covariance converges to the actual covariance, in the sense of Frobenius norm, making fusion without covariance, possible.

*To my Wife, and my Family.*

# Acknowledgements

This thesis has been a result of profound and unwavering support from my supervisor Prof. T. Kirubarajan. I would like to use this opportunity to express my greatest appreciation towards him, for offering me a place in PhD program, for supporting me financially and emotionally, and for all academic and philosophical suggestions throughout this voyage. It has been and always will be a matter of pride that I'm his student.

This thesis would not have completed without the continuous support of Dr. R. Tharmarasa. His patience in dealing with mundane problems coupled with a deep knowledge of the subject has been a strong milestone in my path of attempting independent research. His ways of teaching and engaging with students has left a deep impact in my mind which will continue to persist towards my future goals in academia. I would also thank Dr. Aleksander Jeremic for his valuable advice and suggestions throughout which led to a quality research.

I would like to thank our administrative assistant, Ms. Cheryl Gies for her outstanding and prompt administrative support even during COVID-19. I also appreciate Ms. Tracey Coop for her efforts on the administrative front.

A big thanks to my wife, Gargie who has been by my side in so many ups and downs of this tumultuous journey. I would have quit midway without her love and

support. Thanks to my brother, Akhil and sister, Priti for their consistent and unconditional love throughout these years.

Finally, this thesis is dedicated to my Parents. They have almost sacrificed their entire lives so I can reach here and write this thesis. Thanks for the motivation, Maa and Papa, this is for you.



# Contents

<b>Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>vi</b>
<b>Declaration of Academic Achievement</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Multi-Sensor Network Architectures . . . . .	2
1.2 Track Level Fusion & Thesis Objectives . . . . .	5
1.3 List of Publications . . . . .	12
<b>2 On Conservative Track Fusion Strategies : Harmonic Mean Density</b>	<b>14</b>
2.1 Abstract . . . . .	14
2.2 Introduction . . . . .	15
2.3 Problem Formulation . . . . .	21
2.4 Review of Arithmetic and Geometric Mean Density . . . . .	24
2.5 Proposed Harmonic Mean Density . . . . .	31
2.6 Implementation . . . . .	43
2.7 Simulation . . . . .	53

2.8	Conclusion . . . . .	64
<b>3</b>	<b>Harmonic Mean Density Fusion : Performance and Comparison</b>	<b>67</b>
3.1	Abstract . . . . .	67
3.2	Introduction . . . . .	68
3.3	Harmonic Mean Density Fusion . . . . .	72
3.4	Consistency Analysis of HMD-GA . . . . .	80
3.5	Generalized Mean Density Fusion using Importance Sampling. . . . .	93
3.6	Simulations . . . . .	100
3.7	Conclusion . . . . .	124
<b>4</b>	<b>Multi-Sensor Multi-Target Tracking With Missing Local Covariance</b>	<b>126</b>
4.1	Abstract . . . . .	126
4.2	Introduction . . . . .	127
4.3	Problem Formulation and Fusion Review . . . . .	130
4.4	Converted Measurement Kalman Filter (CMKF) . . . . .	134
4.5	Covariance Initialization at the Fusion Center . . . . .	137
4.6	Generalizing Covariance Initialization . . . . .	149
4.7	Covariance Recursion at Fusion Center . . . . .	152
4.8	Simulation . . . . .	161
4.9	Conclusion . . . . .	183
<b>5</b>	<b>Trigonometric Moments of a Generalized von Mises Distribution And Their Application in 2-D Range-Only Tracking</b>	<b>185</b>
5.1	Abstract . . . . .	185
5.2	Introduction . . . . .	186

5.3	Problem Formulation . . . . .	189
5.4	Update Equations for Range-Only Tracking . . . . .	192
5.5	Conditional Azimuth Density . . . . .	193
5.6	Expectations of Conditional Azimuth Density . . . . .	196
5.7	Deterministic Sampling of Conditional Azimuth Density and Applica- tion to Tracking . . . . .	201
5.8	Simulation Results . . . . .	205
5.9	Conclusion . . . . .	214
<b>6</b>	<b>Conclusion</b>	<b>215</b>
<b>A</b>	<b>Validity of Gaussian density division in (2.6.5)</b>	<b>220</b>
<b>B</b>	<b>Inequality in covariances of HMD-GA and ICI</b>	<b>222</b>
<b>C</b>	<b>Calculation of Fudge Factor in equation (4.5.5)</b>	<b>224</b>
<b>D</b>	<b>Second Order Linearization of CMM in equation (4.6.8)</b>	<b>226</b>
<b>E</b>	<b>Proof of Eqns. (5.6.4) and (5.6.5)</b>	<b>229</b>

# List of Figures

1.1	Different sensor network topologies. . . . .	4
1.2	An illustration of generalized track fusion scenario. $x$ and $\Sigma$ indicates the state and corresponding covariance. $z$ are the measurements modeled by possibly non-linear functions $h(\cdot)$ or $g(\cdot)$ . $b$ is the optional sensor bias and $v$ is the measurement noise. Superscripts indicate either sensor node or target indices, subscripts indicate time. . . . .	6
2.1	Track fusion strategies . . . . .	17
2.2	A non-normalized Gaussian mixture raised to a non-integer power $w \in [0, 1]$ . The original mixture is shown with dashed line, as $\omega$ approaches 0, the covariance increases. . . . .	28
2.3	Normalization constant for various values of $\omega$ in case of GMD (red) and HMD (green). . . . .	37
2.4	Independent Dataset . . . . .	49
2.5	Non-independent Dataset [48] . . . . .	49
2.6	Entropy of the fused density using various methods for a correlation coefficient of $\rho = 0.5$ among local densities. . . . .	51
2.7	Fused density using various methods for a correlation coefficient of $\rho = 0.5$ among local densities. . . . .	52

2.8	(a) Average run time v/s dimension of the Gaussian density and, (b) average run time v/s number of components in a Gaussian mixture. . . . .	53
2.9	Scenario 1 . . . . .	54
2.10	Root mean square error (RMSE) performance : Scenario 1. . . . .	58
2.11	Normalized estimation error squared (NEES) for scenario 1. The HMD can be seen to be fairly consistent along with other strategies. . . . .	58
2.12	Information flow between fusion center and local trackers for scenario 2. . . . .	60
2.13	Scenario 2. . . . .	64
2.14	Root mean square error performance : Scenario 2 . . . . .	64
2.15	Fused tracks with various approaches for scenario 2. The arithmetic mean density has been excluded due to very high track-loss. . . . .	65
3.1	Independent Dataset . . . . .	87
3.2	Non-independent Dataset [48] . . . . .	87
3.3	Uncertainty region (86.5%) for HMD and GMD fusion of Gaussian densities using proposed sampling technique. . . . .	99
3.4	Fusion of Gaussian mixtures using proposed sampling technique. . . . .	100
3.5	Comparison of fusion weights in case of scalar estimates. . . . .	105
3.6	Scenarios : Consistency based tests. . . . .	108
3.7	Fusion result for consistency-based test 1. . . . .	108
3.8	Fusion result for consistency-based test 2. . . . .	108
3.10	Target 1. . . . .	118
3.11	Target 2. . . . .	118
3.12	Target 3. . . . .	118
3.13	Target 4. . . . .	119

3.14	Target 5. . . . .	119
3.15	Target 6. . . . .	119
3.16	Target 7. . . . .	120
3.17	Target 8. . . . .	120
3.18	Target 9. . . . .	120
3.19	Target 10. . . . .	121
3.20	Target 11. . . . .	121
3.21	Target 12. . . . .	121
3.22	Target 13. . . . .	122
3.23	Target 14. . . . .	122
3.24	Target 15. . . . .	122
3.25	Target 16. . . . .	123
3.26	Target 17. . . . .	123
3.27	Target 19. . . . .	123
3.28	Target 20. . . . .	124
4.1	Normalized Error Squared (NES) based on $10^3$ Monte-Carlo iterations.	140
4.2	Frobenius norm of exact covariance in equation (4.7.13) and one-step covariance equation in (4.7.11) v/s process noise intensity, $\tilde{q}$ for an arbitrary linear system. . . . .	160
4.3	True v/s reconstructed NEES for different values of initial range R, when local sensor is running EKF. . . . .	164
4.4	True v/s reconstructed NEES for different values of initial range R, when local sensor is running CMKF. . . . .	164
4.5	Average Frobenius Distance v/s time. . . . .	165

4.6	Mean deviation of range. . . . .	165
4.7	Mean deviation of velocity. . . . .	166
4.8	Scenario 2 – Comparison of truth and estimated tracks. . . . .	170
4.9	Scenario 2 – Measurements v/s Clutter. . . . .	170
4.10	Scenario 2 – Target ID 1. . . . .	171
4.11	Scenario 2 – Target ID 2. . . . .	171
4.12	Scenario 2 – Target ID 3. . . . .	171
4.13	Scenario 2 – Target ID 4. . . . .	171
4.14	Scenario 2 – Target ID 5. . . . .	172
4.15	Scenario 2 – Target ID 6. . . . .	172
4.16	Scenario 2 – Target ID 7. . . . .	172
4.17	Scenario 2 – Target ID 8. . . . .	172
4.18	Scenario 2 – Target ID 9. . . . .	173
4.19	Scenario 2 – Target ID 10. . . . .	173
4.20	Scenario 2 – Target ID 11. . . . .	173
4.21	Scenario 2 – Target ID 12. . . . .	173
4.22	Scenario 2 – Target ID 13. . . . .	174
4.23	Scenario 2 – Target ID 14. . . . .	174
4.24	Scenario 2 – Target ID 15. . . . .	174
4.25	Scenario 2 – Target ID 16. . . . .	174
4.26	Scenario 2 – Target ID 17. . . . .	175
4.27	Scenario 2 – Target ID 19. . . . .	175
4.28	Scenario 2 – Target ID 20. . . . .	175

5.1	(Top) Actual Density, (Middle) Deterministic Sampling using proposed method, (Bottom) Using the technique specified in [58]; for various values of $q$ . . . . .	206
5.2	Computation of integral in eqn. (5.2.1a) for various values of moments of order $m$ . . . . .	207
5.3	Computation of integral in eqn. (5.2.1b) for various values of $m$ . . . .	207
5.4	Error with respect to quadrature based computation. . . . .	209
5.5	Impact of Matlab's variable precision-arithmetic (VPA) on time-performance.	209
5.6	Target-Tracking Simulation Based on Proposed Approach. . . . .	212



# List of Tables

1.1	Comparison between sensor network architectures. . . . .	3
2.1	Parameters : Scenario 1 . . . . .	56
2.2	Parameters : Scenario 2 . . . . .	63
3.1	Relative computational time of fusion algorithms. . . . .	102
3.2	Simulation parameters – Scenario 2 . . . . .	110
3.3	Locative Parameters – Scenario 2 . . . . .	111
4.1	Elements of an initialized track covariance matrix. $R_{ij}$ refers to corresponding element in sensor covariance $\mathbf{R}$ . . . . .	144
4.2	Scenario 1 – Simulation parameters . . . . .	169
4.3	Simulation parameters – Scenario 2 . . . . .	176
5.1	Error in computation ( $m = 1$ ). . . . .	208
5.2	Simulation Parameters . . . . .	210

# Declaration of Academic Achievement

This research presents analytical and computational work carried out solely by Nikhil Sharma, herein referred to as “the author”, with advice and guidance provided by the academic supervisor Prof. T. Kirubarajan. Information that is presented from out-side sources which has been used towards analysis or discussion, has been cited when appropriate, all other materials are the sole work of the author.

# Chapter 1

## Introduction

Target tracking has been an extensively researched area since the last 50 years, when Rudolf E. Kálmán developed the celebrated Kalman filter [14]. Since its inception, target-tracking separated from the umbrella of control systems as a branch, and started developing as a research area of its own. It encompasses various fields such as, probability, statistics, linear systems, operational research, random finite sets, game theory and signal processing among others.

The target-tracking research now is so vast, that a deep dive into all the topics is close to impossible! In the contemporary literature, tracking has been subdivided into several sections — filtering, target (read track) management, sensor-management, networked tracking, association, clutter management, anomaly detection and adversary prevention. All these areas combine themselves to produce a single aim, i.e., given the measurements (or a set of measurements), track the target(s) as optimally as possible. The word ‘optimally’ often has many variations, but for now, let’s assume that we read it ‘as close as possible’ to the target(s), in some sense.

Therefore, the aim is to increase the accuracy of the target-tracking algorithms.

One way of achieving this is to employ the most accurate sensor, implement the best possible tracking algorithm, and hope that the target(s) remain in the field-of-view (FoV) of the sensor at all times. It's obvious to worry about the impracticality of this approach. The sensor is prone to fail, the 'best possible' tracking algorithm is often computationally intensive, and above all, even if we stop contemplating these issue, we need to guarantee that the observability of the target-sensor model is guaranteed. To avoid such problems, implementing a networked, or a multi-sensor tracking systems is very useful.

An example here would assert this statement. Imagine a single target and a network of three static sensors, each measuring range, azimuth and elevation of the target respectively. On their own, the sensors are incapable of estimating the target's position in three dimensions (3D) because of unobservability. But collectively, if they can send their measurements to a collection agent which does the Kalman filter processing for all of them, the system becomes observable and tracking is guaranteed. Appending more such sensors to the network would increase the overall tracking accuracy and also the detection area by multifold, which could be impossible for a single sensor system. This is a classic example of a centralized scenario.


## 1.1 Multi-Sensor Network Architectures

Fig. 1.1 shows a schematic of different sensor network topologies that exist in the target tracking literature. The node  $\bullet$  symbolizes sensors without any processing capabilities, transmitting raw measurements to a centralized fusion center (CFC), which is represented as  $\bullet$ . Another topology is the distributed architecture, wherein, the sensors represented by  $\circ$  are equipped with some processing mechanism such that

Table 1.1: Comparison between sensor network architectures.

	Centralized	Distributed	Decentralized
<b>Optimal</b>	Yes.	No! <sup>1</sup>	No! <sup>1</sup>
<b>Computation load</b>	Medium.	Medium.	High.
<b>Communication load</b>	Heavy.	Minimal.	Minimal.
<b>Analytical Scalability</b>	For Gaussian mixtures.	No process noise.	Measurement consensus
<b>Fault-tolerance</b>	Minimal.	Yes.	Highly scalable.
<b>Major Issues</b>	Minimal.	Yes.	High fault-tolerance.
<b>Dissimilar sensor support</b>	Communication load.	Cross-correlation.	Sub-optimality.
<b>Local observability</b>	Yes.	No! <sup>1</sup>	No! <sup>1</sup>
	Not required.	Not required.	Not required.

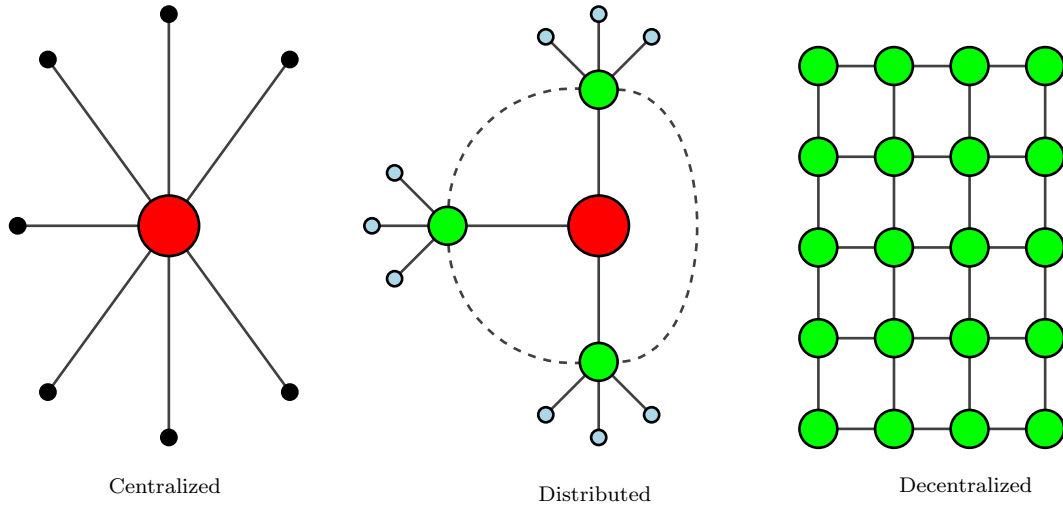
<sup>1</sup> In general.

updated estimates are transmitted to a local fusion center (LFC), shown as . The local fusion centers, in turn, send collective track reports to a central fusion center. Note that the LFCs may or may not be directly connected, which is shown by a dashed line.

The right-most figure in Fig. 1.1 shows a decentralized scheme, wherein the sensors themselves are LFCs that propagate the measurements or processed tracks to their neighboring nodes. Therefore, in this situation, the requirement for a fusion center is nullified. Also, unlike shown in the figure, the decentralized network may not be fully connected.

As one moves from left to right in Fig. 1.1, it is apparent that the computational burden at CFC is gradually transferred to sensor nodes, or LFCs. The current technology ensures the availability of sensors with some processing capabilities. Albeit, one can find an entire tracking system embedded on a microprocessor the size of a thumb. So, it is advantageous to shift the load from a single CFC and distribute it to a number of nodes in its vicinity.

A comparison of the various architectures is shown in Table 1.1. The reader can anticipate that each network architecture has its own pros and cons. A centralized mechanism benefits from its optimality in the sense of minimum mean square error



Different sensor network topologies.

(M.M.S.E.) and the fact that heterogeneous sensor models can be easily integrated. But it also lacks the ability to scale and be fault-tolerant. Following are some of the cons of using a centralized mechanism :

- The sensor nodes transmit raw detections directly to the fusion center. In the event of measurement clutter, this could return a huge number of measurements in a single frame from a single sensor. Hundreds of sensors simultaneously transmitting such information adversely impact the communication links.
- The measurement and the state equations are usually modeled in heterogeneous state space. Therefore, measurement-to-track associations are non-trivial.
- Even though optimality can be achieved, it only occurs in the case of linear Gaussian-Markov systems.
- The centralized system breaks down if the CFC is impacted or compromised

by an adversary. Also, due to the availability of a limited number of physical ports, only a finite number of sensors can be connected.

From Table 1.1, it is evident that distributed fusion is a viable choice and acts as a legit compromise between computation load, communication load, optimality, fault tolerance and scalability. The distributed architecture also emerges when two centralized fusion centers are connected together such that their resulting estimates are fused by a third fusion node. This scheme is also known by the term track-level fusion or track-to-track fusion (T2TF) since processed estimates (which are technically called ‘tracks’) are fused together at the fusion center (FC).

## 1.2 Track Level Fusion & Thesis Objectives

We assume that there are  $T$  targets traversing with discrete-time dynamics represented by the linear state-space model,

$$\mathbf{x}_k^t = \mathbf{F}_{k,k-1}^t \mathbf{x}_{k-1}^t + \mathbf{w}_{k,k-1}^t, \quad (1.2.1)$$

where  $\mathbf{F}_{k,k-1}^t$  is the state transition matrix from time instant  $k-1$  to  $k$ .  $\mathbf{w}_{k,k-1}^t$  is the noise present in the motion model (assumed Gaussian) and  $t = 1, \dots, T$ . Suppose there are  $N$  local nodes, equipped with sensors and processing equipment extracting linear measurements –  $\mathbf{z}_k^i$  at time  $k$  for objects in their vicinity. Here,  $i = 1, \dots, N$  is the node index

$$\mathbf{z}_k^i = \mathbf{H}_k^i \mathbf{x}_k + \mathbf{v}_k^i \quad (1.2.2)$$

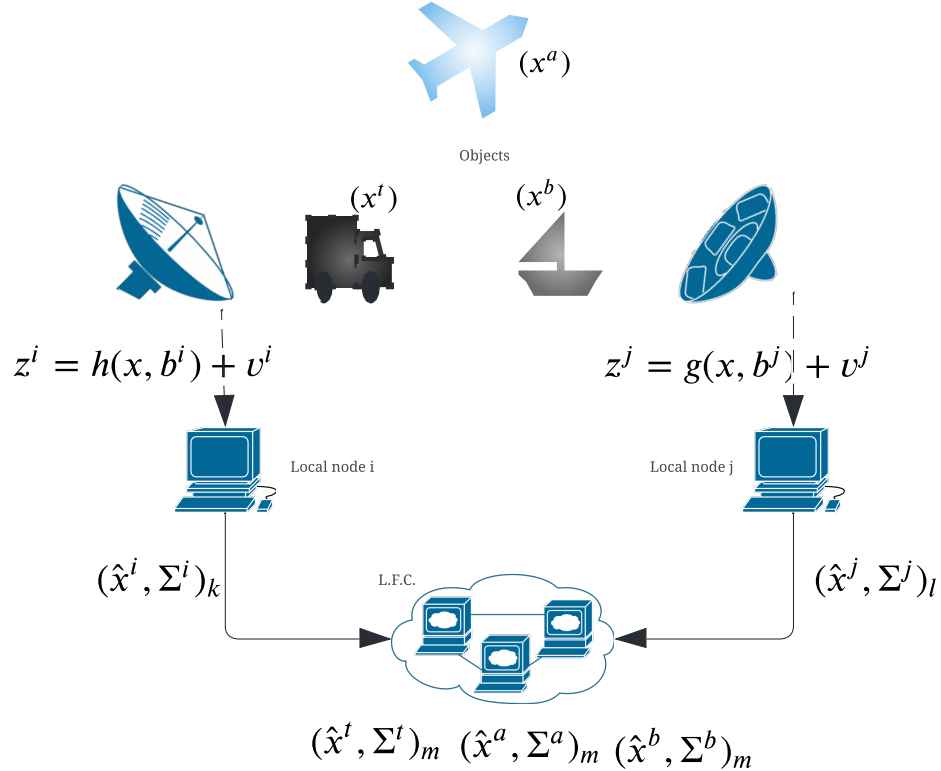


Figure 1.2: An illustration of generalized track fusion scenario.  $x$  and  $\Sigma$  indicates the state and corresponding covariance.  $z$  are the measurements modeled by possibly non-linear functions  $h(\cdot)$  or  $g(\cdot)$ .  $b$  is the optional sensor bias and  $v$  is the measurement noise. Superscripts indicate either sensor node or target indices, subscripts indicate time.

After selecting distributed fusion as the primary focus of this thesis, we assume that each node produces a posterior estimate  $\hat{\mathbf{x}}_{k|k}^{i,t}$  and covariance  $\mathbf{\Gamma}_{k|k}^{i,t}$  according to its measurements and transmits the same to the CFC. The CFC, then, uses these estimates and attempts to fuse them optimally. Since there are multiple tracks, the local tracks need to be associated first. This is, however, exempt from the focus of this thesis. We assume that either an association has already taken place or there is a known single target. Due to this reason, the target index is often exempt from the notation.



### 1.2.1 Thesis Objective

The objective of this thesis is to tackle a fundamental problem in track-level fusion, which is the unknown cross-correlation between local estimates. This inhibits the use of *easy-to-implement* Kalman filter update at the fusion center.

The fusion of tracks can be visualized as the fusion of two probability densities, which represent the spatial distribution of a target at a particular instant. This is similar to a standard Kalman filter, where the prior distribution of the target is fused with the distribution of likelihood. In the case of linear Gaussian-Markov systems, this results in an estimate that is optimal in the sense of maximizing the likelihood as well as minimizing the mean-square error.

One critical assumption to reach this optimality is the independence of measurement noise and process noise, due to which the resulting prior distribution and the measurement likelihood are independent. The Bayes theorem can then be applied, resulting in an updated estimate, which forms the crux of the Kalman filter. Unfortunately, in the case of track-fusion, the local densities are non-independent when they belong to the same target. This occurs primarily due to three reasons :

- If the process noise varies, the estimation error corresponding to the local estimates for that target will also change accordingly. This indicates the presence of a positive correlation between the estimation errors of the same target.
- Correlation also exists due to duplicate measurement sequences. Suppose the global estimate at time  $k$  is updated with the local density  $p^i(\mathbf{x}_k|\mathbf{z}_{1:k})$  at node  $i$ . Then, at time  $k + 1$ , when the local density from the same node,  $p^i(\mathbf{x}_{k+1}|\mathbf{z}_{1:k+1})$  is fused, the global estimate already contains information from the measurement sequence  $\mathbf{z}_{1:k}$ . The presence of duplicate information leads to correlated errors.

- If the local tracks are initialized with the same prior information, the tracks are definitely correlated. This can be explained as follows.

Suppose the local nodes  $i$  and  $j$  initialize the estimate by sampling from the same prior density,  $p_0 \sim \mathcal{N}(\bar{\mathbf{x}}_0; \mathbf{\Gamma})$  where  $\mathbf{\Gamma}$  is the covariance matrix.

$$\hat{\mathbf{x}}_{0|0}^i = \bar{\mathbf{x}}_0 + \mathbf{w}_0^i \quad (1.2.3)$$

$$\hat{\mathbf{x}}_{0|0}^j = \bar{\mathbf{x}}_0 + \mathbf{w}_0^j \quad (1.2.4)$$

It can be seen that even though the samples  $\mathbf{w}_0^i$  and  $\mathbf{w}_0^j$  are independently drawn, the estimates  $\hat{\mathbf{x}}_{0|0}^i$  and  $\hat{\mathbf{x}}_{0|0}^j$  are still correlated.

## 1.2.2 Optimal Track-Fusion Algorithm

If the local tracks  $\hat{\mathbf{x}}_{k|k}^1, \hat{\mathbf{x}}_{k|k}^2, \mathbf{\Gamma}_{k|k}^1, \mathbf{\Gamma}_{k|k}^2$  along with the cross-covariance  $\mathbf{\Gamma}_{k|k}^{12} = \left(\mathbf{\Gamma}_{k|k}^{21}\right)^T$  are available at FC and assumed synchronized, then the optimal algorithm for track fusion is the straight-forward equation,

$$\hat{\mathbf{x}}_{k|k}^f = \hat{\mathbf{x}}_{k|k}^1 + \mathbf{K}_{12} \left[ \hat{\mathbf{x}}_{k|k}^1 - \hat{\mathbf{x}}_{k|k}^2 \right] \quad (1.2.5a)$$

$$\mathbf{\Gamma}_{k|k}^f = \mathbf{\Gamma}_{k|k}^1 - \mathbf{K}_{12} \mathbf{\Gamma}_{k|k}^1 \quad (1.2.5b)$$

where,

$$\mathbf{K}_{12} = \left[ \mathbf{\Gamma}_{k|k}^1 - \mathbf{\Gamma}_{k|k}^{12} \right] \left[ \mathbf{\Gamma}_{k|k}^1 + \mathbf{\Gamma}_{k|k}^2 - \mathbf{\Gamma}_{k|k}^{12} - \mathbf{\Gamma}_{k|k}^{21} \right]^{-1} \quad (1.2.6)$$

The equations resemble the KF equations where the cross-correlation between measurement and process noise has been accounted for, using  $\mathbf{\Gamma}_{k|k}^{12}$ . The problem is to

calculate the  $\mathbf{\Gamma}_{k|k}^{12}$ . This can be done recursively by noting that at time  $(l + 1)$ , the local node estimate error,  $\tilde{\mathbf{x}}_{l+1|l+1}$  can be calculated as,

$$\begin{aligned} \tilde{\mathbf{x}}_{l+1|l+1} &= [\mathbf{I} - \mathbf{K}_{l+1}\mathbf{H}_{l+1}] \mathbf{F}_l \tilde{\mathbf{x}}_{l|l} \\ &\quad - [\mathbf{I} - \mathbf{K}_{l+1}\mathbf{H}_{l+1}] \mathbf{w}_l + \mathbf{K}_{l+1}\mathbf{v}_{l+1} \end{aligned} \quad (1.2.7)$$

where,  $\mathbf{K}_{l+1}$  is the Kalman gain at time instant  $(l + 1)$  and rest of the terms are mentioned in (1.2.1) and (1.2.2). Using the same recursion sequentially, one can arrive at the local node estimate error at time  $(l + 2)$  which is,

$$\begin{aligned} \tilde{\mathbf{x}}_{l+2|l+2} &= [\mathbf{I} - \mathbf{K}_{l+2}\mathbf{H}_{l+2}] \mathbf{F}_{l+1} [\mathbf{I} - \mathbf{K}_{l+1}\mathbf{H}_{l+1}] \mathbf{F}_l \tilde{\mathbf{x}}_{l|l} \\ &\quad - [\mathbf{I} - \mathbf{K}_{l+2}\mathbf{H}_{l+2}] \mathbf{F}_{l+1} [\mathbf{I} - \mathbf{K}_{l+1}\mathbf{H}_{l+1}] \mathbf{w}_l - [\mathbf{I} - \mathbf{K}_{l+2}\mathbf{H}_{l+2}] \mathbf{w}_{l+1} \\ &\quad + [\mathbf{I} - \mathbf{K}_{l+2}\mathbf{H}_{l+2}] \mathbf{F}_{l+1} \mathbf{K}_{l+1}\mathbf{v}_{l+1} + \mathbf{K}_{l+2}\mathbf{v}_{l+2}. \end{aligned} \quad (1.2.8)$$

Proceeding in this fashion, we can calculate the estimate error at time  $k$  given that the last estimate transmitted to F.C. was at time instant  $l$ ,

$$\tilde{\mathbf{x}}_{k|k} = \mathbf{W}_{e_{k,l}}^1 \tilde{\mathbf{x}}_{l|l} + \sum_{i=l+1}^k \mathbf{W}_{wk,i-1}^1 \mathbf{w}_{i-1} + \sum_{i=l+1}^k \sum_{i=l+1}^k \mathbf{W}_{vk,i}^1 \mathbf{v}_i, \quad (1.2.9)$$

for all participating nodes (see [90] for the definition of weights). Since the process and measurement noise are assumed to be white, the cross-correlation  $\mathbf{\Gamma}_{k|k}^{12}$ , given that the cross-correlation at  $l$  is available, is given by,

$$\mathbf{\Gamma}_{k|k}^{12} = \mathbf{W}_{e_{k,l}}^1 \mathbf{\Gamma}_{l|l}^{12} \mathbf{W}_{e_{k,l}}^2 + \sum_{i=l+1}^k \mathbf{W}_{e_{k,l}}^1 \mathbf{Q}_{i-1} \mathbf{W}_{e_{k,l}}^2, \quad (1.2.10)$$

where  $\mathbf{Q}_i$  is the process noise covariance at time instant  $i$ .

**Comments** This is the general solution for non-full rate communication<sup>1</sup>, note that the availability of all Kalman gain matrices from  $l+1$  to  $k$  for each instance is required. The other parameters are needed only once for time-invariant systems.

It was shown in [90] that since the weights  $\mathbf{W}_{ek,l}^i$ ,  $\mathbf{W}_{ek,l}^i$  depend on the Kalman gain and the measurement matrix, which in turn are dependent on the target position estimates and sensor location. These may be reconstructed (even for non-linear systems) approximately at the FC, which avoids the transmission of redundant information from each local node. But due to the approximations in reconstruction, the algorithm is still rendered suboptimal. Therefore, expecting an optimal solution in such a scenario is impossible. The user is forced to use approximations to arrive at cross-correlations or the reconstructed Kalman gain matrix (which requires linearity).

The existing fusion techniques often work around such approximations to arrive at a suboptimal solution. They either require a list of Kalman gain matrices since the last fusion update or the need to store each local covariance at the fusion center (information matrix fusion). Even then, the optimality of a centralized filter is not reached. This thesis invents a near-optimal fusion technique that avoids the need to compute cross-correlations, or local Kalman gain, and could still fuse the local densities such that ‘double counting’ of common information is avoided.

---

<sup>1</sup>Full communication rate refers to the case when local estimates are sent as soon as they are updated, resulting in no benefit in communication load.

### 1.2.3 Theme and Contributions of Thesis

In compliance with the terms and regulations imposed by McMaster University, this dissertation has been written in sandwich thesis format by compiling three journal articles. These articles represent independent research performed by the author of this dissertation.

The articles in this thesis are aimed at track-level fusion without knowledge of cross-correlation and local parameters. A summary of the contributions of the thesis is as follows :

- i)* The thesis explores various existing track fusion methodologies, wherein the aim is to produce a consistent fusion mechanism without bookkeeping and requesting extra information from the sensor nodes.
- ii)* A novel track-fusion technique known as the harmonic mean density (HMD) has been derived. The technique runs on the same computation power as existing methods and has been found to be more accurate. This track fusion methodology is rigorously analyzed mathematically to establish itself as a dependable fusion technique.
- iii)* Several results have been derived to establish the proposed technique as a generalized fusion of probability densities.
- iv)* A sampling-based method for fusion of Gaussian mixture densities has been proposed in this thesis. The technique proposed doesn't require the use of a proposal density, as in importance sampling.

- v)* The existing fusion technique, known as the inverse covariance intersection, existed only for fusing Gaussian densities. The technique was extended to Gaussian mixtures.
- vi)* Track fusion in the case of missing covariance and cross-correlation is a non-trivial problem that has no robust solution so far in the literature. A method for reconstruction of local track covariance at the fusion center has been proposed in this thesis.
- vii)* It was proven that the aforementioned covariance reconstruction method was guaranteed to generate positive definite matrices. Using linearization, the method was extended to any observable sensor model.
- viii)* A new formulation for 2D range-only tracking is proposed, which is based on the calculation of exact circular moments of conditional azimuth density.

## 1.3 List of Publications

The thesis is a compilation of the following articles :

### 1.3.1 Journal Papers

- Sharma N., Bhaumik S., Tharmarasa R., Kirubarajan T.; “On Conservative Track Fusion Strategies : Harmonic Mean Density”. Submitted to IEEE Transactions in Aerospace and Electronic Subsystems; October, 2022.
- Sharma N., Tharmarasa R., Kirubarajan T.; “Harmonic Mean Density Fusion : Performance and Comparison”. To be submitted to IEEE Transactions in

Aerospace and Electronic Subsystems; October, 2023.

- Sharma N., Tharmarasa R., Kirubarajan T.; “Multi-Sensor Multi-target Tracking with Missing Local Covariance”. Submitted to IEEE Transactions in Aerospace and Electronic Subsystems; February, 2023.
- Sharma N., Bhaumik S., Tharmarasa R., Kirubarajan T.; “Trigonometric Moments of a Generalized von Mises Distribution And Their Application in 2-D Range-Only Tracking”. To be submitted to IEEE Transactions in Aerospace and Electronic Subsystems; October, 2023.

### 1.3.2 Technical Reports

- Sharma N., Kirubarajan T.; “Track Fusion Algorithms Library”. Estimation, Tracking and Fusion Research Laboratory, McMaster University, Canada, August, 2022 .
- Sharma N., Kirubarajan T.; “Track Fusion Algorithms : Tracklets, Information and Consensus Fusion”. Estimation, Tracking and Fusion Research Laboratory, McMaster University, Canada; December, 2022.

### 1.3.3 Miscellaneous

- Sharma N., Kirubarajan T.; “Cognitive Radio Algorithms for Constrained Wireless Networks”. Estimation, Tracking and Fusion Research Laboratory, McMaster University, Canada; February, 2022.

# Chapter 2

## On Conservative Track Fusion

### Strategies : Harmonic Mean

### Density

#### 2.1 Abstract

In a distributed sensor fusion architecture, naive fusion can lead to degraded results as track correlations are ignored, and conservative fusion strategies are employed as a suboptimal alternative to the problem. Gaussian mixtures provide a flexible means of modeling any density; therefore, fusion strategies suitable for use with Gaussian mixtures are needed. While the generalized covariance intersection (CI) provides a means to fuse Gaussian mixtures, the procedure is cumbersome and requires evaluating a non-integer power of the mixture density. In this chapter, we develop a conservative fusion strategy based on the harmonic mean density (HMD) interpolation of local densities and show that the proposed method can handle both Gaussian and mixture



densities within the same framework. The mathematical properties of the proposed fusion strategy are studied and simulated in 2D and 3D maneuvering target tracking scenarios. The simulation suggests that the proposed HMD fusion performs better than other conservative strategies by posing a tighter conservative bound.

## 2.2 Introduction

A distributed data fusion scenario comprises several sensor platforms connected to a node where the local information from the sensors is fused. Since a sensor can capture multiple detections in the presence of clutter, sending raw measurements is avoided. Instead, the sensor communicates estimated target states (tracks), which is why the term track-to-track fusion (T2TF) is commonly used.

Tracks estimating the same target trajectory in T2TF are never independent, even though the local measurement errors are uncorrelated [9]. The dependence emerges due to the common process noise and temporal correlation between tracks arriving from the same platform. Ignoring this correlation can lead to degraded performance as the estimates are optimistic [12]. The optimism is due to the aggregation of “*common information*” in sensor estimates. This common information has to be accounted for and removed from the fused information to avoid track divergence.

The approaches used in accounting for cross-correlation for the last three decades can be categorized into:

- i) Optimal solution based on evaluating exact recursion for correlation, which can later be accounted for in fusion. This strategy would require extra information from local sensors like Kalman gain, sensor and motion model parameters [9] [25].

- ii)* Conditionally optimal methods like information matrix fusion, which would also require knowledge of previous local estimates [24]. Such methods are optimal only for a full communication rate, i.e., the fusion occurs every time the local estimates are updated.
- iii)* Tracklet and quasi-tracklet fusion methods are based on a track’s decorrelation by marginalizing its joint density.
- iv)* Robust and suboptimal solutions like generalized covariance intersection utilize a geometric mean (GM) or the arithmetic mean (AM) interpolation of individual track densities [47, 60].

Fig. 2.1 represents a pictorial representation of available T2TF algorithms. Generalized covariance intersection/Chernoff fusion/geometric averaging have attracted researchers due to its robustness and stability [15]. An essential aspect of this method is that it essentially constitutes a density function which is a log-linear combination of the individual local posterior densities [47], such that the propagation of the “common information” is avoided [8]. Due to computational intensiveness of geometric mean for Gaussian mixtures, an alternative was proposed in [60, 1].

With the advent of computation and embedded technologies, the use of mixture densities has been adopted to deal with multi-modal, non-symmetric noise distributions and multiple-model approaches to filtering. Moreover, the Gaussian mixture models (GMM) possess several standard results available in closed form. Some examples utilizing this advantage include the interacting multiple model (IMM) filter and its modifications for tracking maneuvering targets [16], the multiple hypothesis tracker (MHT) for multiple target tracking [14], Gaussian sum filter [6] and the probability hypothesis density (PHD) filter [29]. These filters are extensively used owing

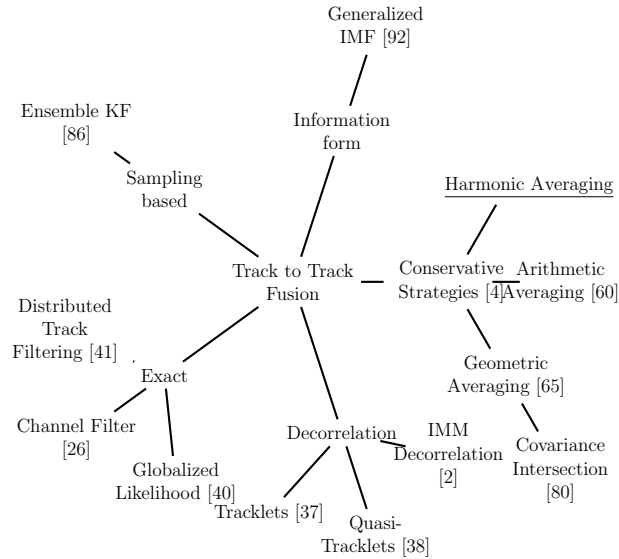


Figure 2.1: Track fusion strategies

to their performance in practical situations. However, the fusion of such mixture-modeled tracks with cross-correlation becomes infeasible due to the unavailability of closed-form expressions. For instance, decorrelation requires the division of prior and posterior densities, for which an exact result is impossible if Gaussian mixtures are involved [3]. Chernoff fusion requires computing non-integer powers of a Gaussian mixture, which does not assume a closed-form [43]. The arithmetic averaging, though feasible, presents a relatively larger uncertainty due to the amalgamation of densities. Therefore, further processing in the form of mode merging and pruning is needed [60]. Due to such reasons, fusion of Gaussian mixtures is non-trivial in practical scenarios, and we need to search in alternative directions. This article proposes a method that can handle uni-modal and multi-modal Gaussian densities without altering the framework.

Research on Gaussian mixture fusion is contemporary, and various solutions are available in the literature. [72] addressed the issue of deriving the dependency relation

between Gaussian mixture estimates and found that the dependency is itself a Gaussian mixture. No remarks on how to ease the fusion, however, were made. A tracklet [38] based fusion strategy for multiple model trackers is proposed in [2], wherein the information decorrelation approach was extended to Gaussian mixture densities and then naive fusion was applied to the decorrelated mixture. This work required division by a Gaussian mixture, wherein a crude approximation of component-wise division was applied. In [100], the dynamic state of an IMM running fusion center was derived when the local tracks are in turn IMM using the “inside information”. But, this approach was based on the assumption that individual motion models in the IMM are entirely the same, except for the process noise. Thus, the approach is not useful in scenarios where, for example, nearly-constant velocity (NCV) and nearly-constant acceleration (NCA) models are employed in parallel.

Recently, conservative fusion methodology has taken a different direction, where the use of Fréchet-means [70] has been explored. These studies are based on the fact that averaging rules are robust to double counting, thus imparting robustness with respect to the correlations involved without requiring any extra information from the local trackers. In [4], a formal definition of conservativeness was presented and proved for arithmetic and geometric averaging. The definition was based on the amount of uncertainty in the distribution, thus using measures such as entropy and Kullback-Leibler divergence. In [60] the arithmetic averaging was applied to a multi-target tracking scenario using the multi-Bernoulli process, in which each component was represented by a Gaussian mixture. In [8], the notion of conservativeness was examined using the notion of entropy. It was proven that geometric mean fusion increases the entropy for positive values of the exponent, indirectly proving conservativeness.

The generalized framework for Chernoff fusion was proposed in [65] by Mahler as the generalized Uhlmann-Julier covariance intersection. Also, in [47], Hurley succinctly expressed the relation between Chernoff information and covariance intersection. In [15], the Chernoff fusion was studied for a non-Gaussian distribution and was termed the geometric mean density (GMD). Since the covariance intersection (CI) assumes a closed form for Gaussian densities, component-wise CI can be used while fusing Gaussian mixtures using Chernoff information. This was the approach used in [99], where the GMM were fused for a bearing-only tracking scenario. A better result was discussed in [49], where a first-order approximation of the power of a Gaussian mixture was employed. This was the first paper that dealt with approximating the power of a Gaussian mixture model with another Gaussian mixture. The result was succeeded by Gunay et al., which employed sigma-points for generalizing the power of a Gaussian mixture [43] but it was computationally demanding. Moreover, such a fusion strategy is affected adversely when the individual modes of a mixture are very close, which is usually the case with an IMM tracker. A review of various methods for approximating the non-integer power of a Gaussian mixture was presented in [5] along with a comparison.

This chapter presents the development of a new mean-based fusion strategy that can deal with Gaussian mixtures with minimal approximation. For Gaussian mixture densities, the common information takes the form of a mixture density based on the fusion weights ( $\omega$ ). We approximate the common information as Gaussian and plug it into the harmonic mean formulation. The resulting density is available in closed-form. In terms of computation, the approach performs similarly to the Chernoff fusion of Gaussian mixtures (which is also the fastest method) and can be shown to be more

accurate.

Other contributions of the chapter are enumerated as follows:

- i)* Detailed investigation of existing average-based fusion algorithms, specifically the arithmetic mean density (AMD) and the geometric mean density (GMD), has been performed. Their issues in implementation while addressing T2TF with Gaussian mixture densities are addressed.
- ii)* The mathematical development of the proposed harmonic mean density based fusion is elucidated along with its various properties in line with the other average based consensus methods.
- iii)* This chapter introduces the solution of fusion weights based on the common information rather than K-L divergence. It has been shown that harmonic fusion simplifies the expression for common information, which could be solved using the expectation-maximization (EM) algorithm.
- iv)* A comparison among fusion strategies based on two real-life target tracking scenarios has been made.

The rest of the chapter is organized as follows: In Section 2.3, a brief summary of the issue of cross-correlation in a distributed sensor fusion scenario is presented. In Section 2.4, a review of the arithmetic and geometric mean fusion along with their implementation for Gaussian mixtures is presented. These include the covariance intersection, pseudo-Chernoff fusion (PCF) for geometric averaging, and the advantageous properties of arithmetic averaging. The proposed harmonic mean density fusion is discussed in Section 2.5, including its conservativeness. Section 2.6 presents

the approximate implementation strategies for the harmonic mean fusion, as it requires division by a Gaussian mixture. The algorithm is simulated on a 2D and 3D scenarios, and the results, along with discussion on the performance of different strategies, are presented in Section 2.7. Finally, the article is concluded in Section 2.8.

## 2.3 Problem Formulation

Considering a scenario with two sensor platforms –  $\mathcal{S}^1$  and  $\mathcal{S}^2$  generating processed track density conditioned on current measurement  $\mathbf{z}_k^i$  —  $p(\mathbf{x}_k|\mathbf{z}_k^i)$ . Where  $k$  is the time step and  $i \in \{1, 2\}$ . Assume that both tracks are conditioned on a common prior density at time  $k$  which is quantified as  $p_c(\mathbf{x}_k)$ . Then,

$$p(\mathbf{x}_k|\mathbf{z}_k^i) \propto p(\mathbf{z}_k^i|\mathbf{x}_k)p_c(\mathbf{x}_k). \quad (2.3.1)$$

Using naive fusion with such local density results in equation (2.3.2)

$$p_N(\mathbf{x}_k|z_k^1 \cup z_k^2) \propto p(\mathbf{z}_k^1|\mathbf{x}_k)p_c(\mathbf{x}_k)p(\mathbf{z}_k^2|\mathbf{x}_k)p_c(\mathbf{x}_k), \quad (2.3.2)$$

where  $p_N(\mathbf{x}_k|z_k^1 \cup z_k^2)$  is the naively fused density. Thus, the common information is accounted for twice, which will lead to an optimistic estimate (e.g., multiplying two Gaussian densities results in a Gaussian density with a lower or the same covariance). A heuristic approach would be to use naive fusion while dividing once with

the common information,

$$\begin{aligned} p(\mathbf{x}_k | \mathbf{z}_k^1 \cup \mathbf{z}_k^2) &\propto \frac{p(\mathbf{z}_k^1 | \mathbf{x}_k) p_c(\mathbf{x}_k) p(\mathbf{z}_k^2 | \mathbf{x}_k) p_c(\mathbf{x}_k)}{p_c(\mathbf{x}_k)}, \\ &\propto \frac{p(\mathbf{x}_k | \mathbf{z}_k^1) p(\mathbf{x}_k | \mathbf{z}_k^2)}{p(\mathbf{x}_k | \mathbf{z}_k^1 \cap \mathbf{z}_k^2)}. \end{aligned} \quad (2.3.3)$$

This is also the exact Bayesian formulation for distributed data fusion [64], where  $p(\mathbf{x}_k | z_k^1 \cap z_k^2)$  accounts for the density due to common prior. For Gaussian densities, an exact formulation of the common prior was presented in [9], which will require extra information from local trackers. This chapter focuses on a class of solutions where the common information can be accounted for (in a suboptimal sense) by only using the information available. Thus,

$$p(\mathbf{x}_k | \mathbf{z}_k^1 \cap \mathbf{z}_k^2) \propto \mathcal{F}(p(\mathbf{x}_k | \mathbf{z}_k^1), p(\mathbf{x}_k | \mathbf{z}_k^2), \omega), \quad (2.3.4)$$

where,  $\mathcal{F} : \mathbb{R}^n \rightarrow \mathbb{R}$  is an appropriate function, and  $\omega \in \{0, 1\}$  is a parameter that can be evaluated to result in the least possible conservative density than that of equation (2.3.3) which is the optimal Bayes' fusion.

If the local platform employs an IMM tracker, then the posterior density  $p(\mathbf{x}_k | z_k^i)$  consists of estimates from multiple modes packed in a mixture (usually Gaussian) of the form,

$$p(\mathbf{x}_k | \mathbf{z}_k^i) = \sum_{m=1}^M \mu^m \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^m, \mathbf{\Gamma}_k^m), \quad (2.3.5)$$

where  $M$  is the total number of modes and  $\mu^m$  is the mode probability of the  $m^{\text{th}}$  Gaussian component. An exact solution to the fusion of such tracks is infeasible due



to,

- i)* Communication constraint as  $m \times \mathcal{S}$  number of filter gains and measurement Jacobian matrices need to be communicated at each fusion instant.
- ii)* Division by a Gaussian mixture [3].

The second point leads to suboptimality even if exact procedures are followed. This necessitates the search for a robust and suboptimal class of strategies termed conservative fusion [4].

### 2.3.1 Conservative Fusion

An estimate pair  $(\hat{\mathbf{x}}_k, \mathbf{\Gamma}_k)$  is termed conservative if its covariance is greater than the true mean square error (MSE) [4]. Thus,

$$\mathbf{\Gamma}_k \succeq E [(\hat{\mathbf{x}}_k - \mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T], \quad (2.3.6)$$

where  $\succeq$  implies inequality in the sense of positive-definiteness. This definition was extended to probability density functions using entropy [8]. A posterior density  $p$  is conservative with respect to the true density  $p_t$  if,

$$\mathbb{H}(p) - \mathbb{H}(p_t) \geq 0, \quad (2.3.7)$$

and  $p(\mathbf{x}_k | \mathbf{z}_k^1 \cap \mathbf{z}_k^2)$  is fused only once at each  $k$ . Here  $\mathbb{H}(\cdot)$  denotes differential entropy [32]. The entropy property of a conservative density imparts decreasing information certainty with respect to the optimal density, so that the estimates are not overoptimistic. Working with track densities as the only available information is a reasonable

compromise.

Another important property of the conservative class is the preservation of the shape [8]. Shape preservation ensures that the location of relative maxima and minima remains unchanged. This is expected, as the shift of the mean relative to the true mean is undesirable. Thus, the class of densities should follow

$$\nabla_{\mathbf{x}}\{p_t(\mathbf{x})\} = 0 \Leftrightarrow \nabla_{\mathbf{x}}\{p(\mathbf{x})\} = 0, \quad (2.3.8)$$

where  $\nabla_{\mathbf{x}}$  is the gradient operator with respect to  $\mathbf{x}$ . The primary motive in this work is to construct a fused density  $p(\mathbf{x}_k|\mathbf{z}_k^1 \cup \mathbf{z}_k^2)$  from  $p(\mathbf{x}_k|\mathbf{z}_k^1)$  and  $p(\mathbf{x}_k|\mathbf{z}_k^2)$ , which is the least conservative than the optimal fusion strategy in equation (2.3.3), in the sense of equation (2.3.7), such that equation (2.3.8) is satisfied.

## 2.4 Review of Arithmetic and Geometric Mean Density

A brief review of methods involving approximating a conservative density using the abstract means [69] or the generalized quasi-arithmetic mean [68] is discussed in this section. As provided in [69], a mixture density  $\mathbf{M}_\omega\{p_1(\mathbf{x}_k), p_2(\mathbf{x}_k)\}$  is an M-mixture if it generates an interpolation between the densities  $p_1(\mathbf{x}_k)$  and  $p_2(\mathbf{x}_k)$  with respect to an  $\omega$ -weighted mean  $\mathcal{M}$ ,

$$\mathbf{M}_\omega\{p_1(\mathbf{x}_k), p_2(\mathbf{x}_k)\} = \frac{\mathcal{M}_\omega\{p_1(\mathbf{x}_k), p_2(\mathbf{x}_k)\}}{\zeta_{\mathcal{M}}\{p_1(\mathbf{x}_k), p_2(\mathbf{x}_k)\}}, \quad (2.4.1)$$

where the normalization constant  $\zeta_{\mathcal{M}}\{p_1(\mathbf{x}_k), p_2(\mathbf{x}_k)\}$  ensures that the mixture  $\mathbf{M}_{\omega}(\cdot)$  is a valid probability density function. The mean  $\mathcal{M}_{\omega}$  can be generalized to  $n$  number of densities if  $\omega$  belongs to a  $n$ -dimensional probability simplex. Some implementations already existing in the literature are,

$$\mathcal{M}_{\omega}^g\{p_1(\mathbf{x}_k), p_2(\mathbf{x}_k)\} = \{p_1(\mathbf{x}_k)\}^{\omega} \{p_2(\mathbf{x}_k)\}^{1-\omega}, \quad (2.4.2)$$

$$\mathcal{M}_{\omega}^a\{p_1(\mathbf{x}_k), p_2(\mathbf{x}_k)\} = \omega p_1(\mathbf{x}_k) + (1 - \omega) p_2(\mathbf{x}_k). \quad (2.4.3)$$

Equation (2.4.2) is the well known geometric mean [49, 71], whereas equation (2.4.3) assumes the form of the arithmetic average [60, 66].

### 2.4.1 Geometric Mean Density (GMD) Fusion

The geometric mean density (GMD) as proposed in [47, 65] is a fusion strategy that assumes the form,

$$\mathbf{M}_{\omega}^g\{p_1(\mathbf{x}_k), p_2(\mathbf{x}_k)\} = \frac{\{p^1(\mathbf{x}_k)\}^{\omega} \{p^2(\mathbf{x}_k)\}^{1-\omega}}{\int \{p^1(\mathbf{x}_k)\}^{\omega} \{p^2(\mathbf{x}_k)\}^{1-\omega} d\mathbf{x}_k}, \quad (2.4.4)$$

where  $\omega \in [0, 1]$  which is selected by solving a suitable optimization problem [98]. It has been proved in [8, 48] that the fusion strategy in equation (2.4.4) is immune to double counting. An appealing property of equation (2.4.4) is that it admits a closed-form solution for an exponential family of distributions. For instance, if the

densities  $p^1(\mathbf{x})$  and  $p^2(\mathbf{x})$  follow a Gaussian distribution,

$$p^1(\mathbf{x}_k) \sim \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^1, \mathbf{\Gamma}_k^1), \quad (2.4.5a)$$

$$p^2(\mathbf{x}_k) \sim \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^2, \mathbf{\Gamma}_k^2), \quad (2.4.5b)$$

then, the geometric mean density fusion reduces to *covariance intersection*, with the fused mean  $\hat{\mathbf{x}}^f$  and covariance  $\mathbf{\Gamma}^f$ ,

$$\mathbf{\Gamma}_k^f = \left( \omega \mathbf{\Gamma}_k^{1^{-1}} + (1 - \omega) \mathbf{\Gamma}_k^{2^{-1}} \right)^{-1}, \quad (2.4.6a)$$

$$\hat{\mathbf{x}}_k^f = \mathbf{\Gamma}_k^f \left( \omega \mathbf{\Gamma}_k^{1^{-1}} \hat{\mathbf{x}}_k^1 + (1 - \omega) \mathbf{\Gamma}_k^{2^{-1}} \hat{\mathbf{x}}_k^2 \right). \quad (2.4.6b)$$

Though, closed-form solutions for exponential families are readily available, the complexity of using GMD for Gaussian mixtures is a harder problem since a non-integer power of a Gaussian mixture is not always a valid distribution, let alone a closed-form solution (also see Fig. 2.2). The authors in [3] focused solely on methods involving non-integer power of Gaussian mixtures. But, a simple solution termed the pseudo-Chernoff fusion uses the following first-order approximation for the power of a Gaussian mixture,

$$\begin{aligned} [p(\mathbf{x}_k)]^\omega &= \left[ \sum_{m=1}^M \mu^m \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^m, \mathbf{\Gamma}_k^m) \right]^\omega, \\ &\approx \sum_{m=1}^M (\mu^m)^\omega \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^m, \mathbf{\Gamma}_k^m)^\omega, \end{aligned} \quad (2.4.7)$$

and the power of a Gaussian distribution is a scaled Gaussian distribution with the suitable normalization constant, i.e.

$$\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^m, \mathbf{\Gamma}_k^m)^\omega \triangleq \alpha_\omega \mathcal{N}\left(\mathbf{x}_k; \hat{\mathbf{x}}_k^m, \frac{\mathbf{\Gamma}_k^m}{\omega}\right), \quad (2.4.8)$$

where the normalization constant,  $\alpha_\omega$  is,

$$\alpha_\omega = \sqrt{\frac{|2\pi \frac{\mathbf{\Gamma}_k^m}{\omega}|}{|2\pi \mathbf{\Gamma}_k^m|^\omega}}. \quad (2.4.9)$$

Note that this approximation for the power of a Gaussian mixture is good only when the components are far away, and the quality of the approximation deteriorates as the components of the mixture become closer to each other [43]. Due to this reason, the pseudo-Chernoff fusion is not a decent candidate for IMM trackers, especially if modes are based on similar state dynamics.

Another approximation for the power of a Gaussian mixture known as the sigma-point Chernoff fusion (SPCF) was presented in [43], which was based on approximating the powered Gaussian as follows

$$q(\mathbf{x}_k) = \left[ \sum_{m=1}^M \mu^m \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^m, \mathbf{\Gamma}_k^m) \right]^\omega, \quad (2.4.10a)$$

$$\approx \sum_{m=1}^M \beta_k^m \mathcal{N}\left(\mathbf{x}_k; \hat{\mathbf{x}}_k^m, \frac{\mathbf{\Gamma}_k^m}{\omega}\right). \quad (2.4.10b)$$

The approximation is appropriate, as shown in Fig. 2.2. The modes of the resulting un-normalized density are unchanged, but the variance of each component is inflated, which is assumed to be proportional to  $\omega$  in the approximation. The next step is to solve the following optimization problem using deterministic sampling to find the

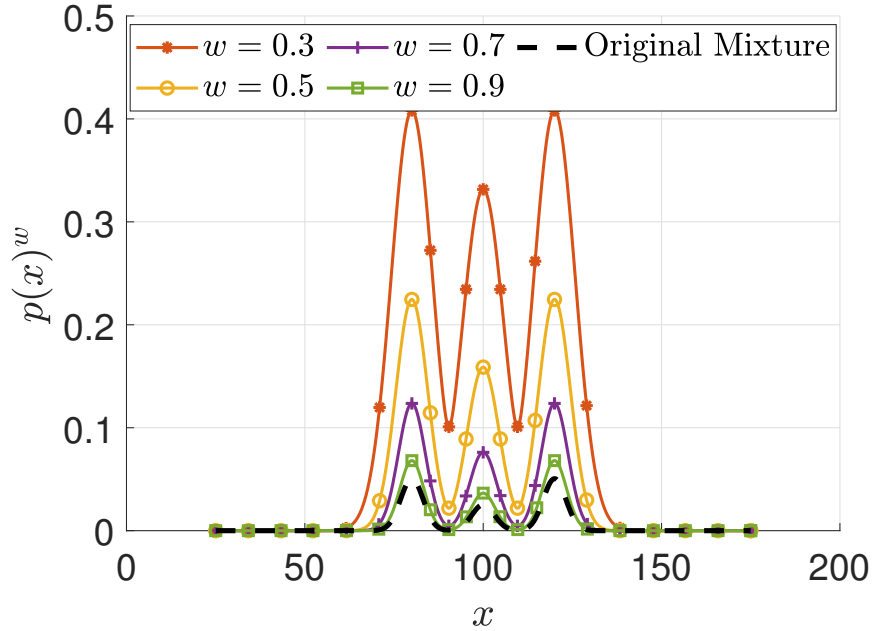


Figure 2.2: A non-normalized Gaussian mixture raised to a non-integer power  $w \in [0, 1]$ . The original mixture is shown with dashed line, as  $\omega$  approaches 0, the covariance increases.

component weights  $\beta_k = \left[ \beta_k^1 \quad \beta_k^2 \quad \dots \quad \beta_k^m \right]^T$ ,

$$\beta_k^* = \arg \min_{\beta_k} \int (q(\mathbf{x}_k) - (p(\mathbf{x}_k))^\omega)^2 p(\mathbf{x}_k) d\mathbf{x}_k \quad (2.4.11a)$$

$$\text{s.t. } \beta_k^m \geq 0, \quad m = 1, \dots, M. \quad (2.4.11b)$$

Thus, prior to the fusion of track densities, it is vital to evaluate sigma-points and solve equation (2.4.11) for each density, which increases the computation of the fusion process drastically. For the example in [43], the sigma-point-based approximation was six times more computationally costly than the approximation in equation (2.4.7). Once the power of the track densities is calculated, one can simply use naive fusion, since the independence has been accounted for using  $\omega$ .

### 2.4.2 Arithmetic Mean Density (AMD) Fusion

The arithmetic mean density is the result of applying the linear fusion approach to density functions. The AMD fusion formula is given by

$$\mathbf{M}_\omega^a \{p_1(\mathbf{x}_k), p_2(\mathbf{x}_k), \dots, p_n(\mathbf{x}_k)\} = \sum_i^N \omega_i p_i(\mathbf{x}_k), \quad (2.4.12)$$

which always results in a valid probability density function as long as  $\sum_i \omega_i = 1$ . Different probability densities can be conveniently fused using the arithmetic average owing to its attractive property of resulting in a closed-form mixture, as demonstrated in [34]. It was also pointed out that AMD also serves as the “minimizing density” for the average K-L divergence  $\mathbb{D}_{KL}$ ,

$$\mathbf{M}_\omega^a(\cdot) = \arg \min_{g(\mathbf{x}_k)} \sum_i \omega_i \mathbb{D}_{KL}(p_i(\mathbf{x}_k) : g(\mathbf{x}_k)), \quad (2.4.13)$$

where  $g(\mathbf{x}_k)$  is the argument of minimization. The above property does not emphasize the quality of the fusion strategy, but can be exploited as an optimization problem that results in  $\omega_i$ . The minimization merely states that the output (fused) density after AMD fusion is at minimum distance, in the sense of K-L divergence, with the participating local densities. In other words, while each local posterior is replaced by a conservative version of itself and then fused, the change it undergoes is minimal. This minimization does not provide any motivation for optimal distributed fusion.

An important property of the arithmetic average is that the fusion approach is consistent even if one of the densities is inconsistent [8]. The GMD in equation (2.4.4) however requires that both probability densities are consistent to output a consistent fused result. The conservativeness of AMD has also been proved in [4].

An implementation of AMD in a multi-sensor, multi-target tracking scenario was performed in [60], where the linear fusion approach was applied as a target-wise fusion rule to a multi-Bernoulli process.

The major problem with AMD fusion is its *over-conservativeness*, which results in a larger variance. It can be seen that since AMD fusion is an aggregating process, it results in a much larger variance than either component. This *increased bandwidth* can be advantageous in situations with biased estimators, which have an offset from the estimator mean. The immunity of AMD to handle offsets and biases can also be extended to cases with misdetection and false alarms, thereby increasing its accuracy in such cases. Another advantage of AMD is its computational efficiency for exact formulation of Gaussian mixtures and particles, enabling real-time processing. Also note that the calculation of the normalization constant is trivial in this case, unlike GMD.

However, as we will see in Section 2.7, the AMD performs poorly in cases involving unobservable local nodes where feedback is required. The large variance of the AMD leads to a more uncertain fused estimate, which usually takes a longer time to settle down. The requirement of clustering, pruning, and merging as a post-processing step in AMD is also a fundamental disadvantage [60].



## 2.5 Proposed Harmonic Mean Density

The generalization of the harmonic mean and their ubiquity is given in [68, 55, 35]. The weighted harmonic mean of two scalar estimates  $\hat{x}^1$  and  $\hat{x}^2$  is given by the expression,

$$H_\omega(\hat{x}_k^1, \hat{x}_k^2) = \frac{\hat{x}_k^1 \hat{x}_k^2}{(1 - \omega)\hat{x}_k^1 + \omega\hat{x}_k^2}. \quad (2.5.1)$$

The harmonic mean can also be obtained from quasi-arithmetic mean using the monotone function  $f(x) = \frac{1}{x}$  [18]. Based on the statistical M-mixture definition given in equation (2.4.1), the harmonic mean interpolated between  $p_1(\mathbf{x}_k)$  and  $p_2(\mathbf{x}_k)$  is,

$$\mathcal{M}_\omega^h\{p_1(\mathbf{x}_k), p_2(\mathbf{x}_k)\} = \frac{p_1(\mathbf{x}_k)p_2(\mathbf{x}_k)}{(1 - \omega)p_1(\mathbf{x}_k) + \omega p_2(\mathbf{x}_k)}, \quad (2.5.2)$$

and the corresponding density,

$$\mathbf{M}_\omega^h\{p_1(\mathbf{x}_k), p_2(\mathbf{x}_k)\} = \frac{1}{\zeta_{\mathcal{M}}^h} \frac{p_1(\mathbf{x}_k)p_2(\mathbf{x}_k)}{(1 - \omega)p_1(\mathbf{x}_k) + \omega p_2(\mathbf{x}_k)}, \quad (2.5.3)$$

where,

$$\zeta_{\mathcal{M}}^h = \int_{\mathbb{R}^n} \frac{p_1(\mathbf{x}_k)p_2(\mathbf{x}_k)}{(1 - \omega)p_1(\mathbf{x}_k) + \omega p_2(\mathbf{x}_k)} d\mathbf{x}_k. \quad (2.5.4)$$

Comparing equation (2.5.4) with the exact Bayesian density in equation (2.3.3), the only difference is that the harmonic averaging accounts for the common information  $p_c(x)$  as a convex combination of the individual densities (precisely, a weighted

arithmetic mean),

$$p(\mathbf{x}_k | \mathbf{z}_k^1 \cap \mathbf{z}_k^2) \propto (1 - \omega)p_1(\mathbf{x}_k | \mathbf{z}_k^1) + \omega p_2(\mathbf{x}_k | \mathbf{z}_k^2). \quad (2.5.5)$$

This is also analogous to the geometric mean density  $\mathcal{M}_\omega^h$  as,

$$\mathcal{M}_\omega^h \propto p_1(\mathbf{x}_k)^\omega p_2(\mathbf{x}_k)^{(1-\omega)} \quad (2.5.6)$$

$$= \frac{p_1(\mathbf{x}_k)p_2(\mathbf{x}_k)}{p_1(\mathbf{x}_k)^{(1-\omega)}p_2(\mathbf{x}_k)^\omega}. \quad (2.5.7)$$

In this case, the common information is approximated by the geometric mean of individual densities. Interestingly, the similarity is shown by the arithmetic average as well, where,

$$p(\mathbf{x}_k | \mathbf{z}_k^1 \cup \mathbf{z}_k^2) = \omega p_1(\mathbf{x}_k) + (1 - \omega)p_2(\mathbf{x}_k) \quad (2.5.8)$$

$$= \frac{p_1(\mathbf{x}_k) \times p_2(\mathbf{x}_k)}{\frac{p_1(\mathbf{x}_k)p_2(\mathbf{x}_k)}{\omega p_1(\mathbf{x}_k) + (1-\omega)p_2(\mathbf{x}_k)}}. \quad (2.5.9)$$

Hence,

$$p(\mathbf{x}_k | \mathbf{z}_k^1 \cap \mathbf{z}_k^2) \propto \frac{p_1(\mathbf{x}_k)p_2(\mathbf{x}_k)}{\omega p_1(\mathbf{x}_k) + (1 - \omega)p_2(\mathbf{x}_k)}. \quad (2.5.10)$$

Thus, to summarize;

- (i) The AMD fusion approximates the common information as the harmonic mean of individual posterior densities.
- (ii) The GMD uses the geometric mean itself as the common information density.

- (iii) The HMD approximates the common information as an arithmetic average of the posterior track densities.

### 2.5.1 Properties of Harmonic Average Density

The conservative properties of the AMD and the GMD have been proved in [5, 8]. The standard inequality states  $AM \geq GM \geq HM$  and the conservativeness if any, could be breached by the harmonic averaging. Thus, it is required to prove the conservativeness of the harmonic interpolation with respect to the true Bayesian density before proceeding with its usage.

**Theorem 1.** *The harmonic fusion avoids double counting of information.*

*Proof.* The proof is similar to that for the other two mean densities [8] where conditional dependence is assumed. Hence, (omitting  $k$ ),

$$p(\mathbf{z}^i|\mathbf{x}) = p(\mathbf{z}^{i/j}|\mathbf{x})p(\mathbf{z}^i \cap \mathbf{z}^j|\mathbf{x}), \quad (2.5.11)$$

where  $p(\mathbf{z}^{i/j}|\mathbf{x})$  denotes the exclusive information present with sensor track  $i$  relative to  $j$  such that  $p(\mathbf{z}^{i/j}|\mathbf{x}) \cap p(\mathbf{z}^{j/i}|\mathbf{x}) = \emptyset$ . Using Bayes' theorem and proceeding with

the assumption above,

$$\begin{aligned}
& p(\mathbf{x}|\mathbf{z}^1 \cup \mathbf{z}^2) \\
& \propto \frac{p(\mathbf{x}|\mathbf{z}^1)p(\mathbf{x}|\mathbf{z}^2)}{(1-\omega)p(\mathbf{x}|\mathbf{z}^1) + \omega p(\mathbf{x}|\mathbf{z}^2)} \\
& \propto \frac{p(\mathbf{z}^1|\mathbf{x})p(\mathbf{z}^2|\mathbf{x})p(\mathbf{x})}{(1-\omega)p(\mathbf{z}^1|\mathbf{x}) + \omega p(\mathbf{z}^2|\mathbf{x})} \\
& = \frac{p(\mathbf{z}^{1/2}|\mathbf{x})p(\mathbf{z}^1 \cap \mathbf{z}^2|\mathbf{x})p(\mathbf{z}^{2/1}|\mathbf{x})p(\mathbf{z}^1 \cap \mathbf{z}^2|\mathbf{x})p(\mathbf{x})}{(1-\omega)p(\mathbf{z}^{1/2}|\mathbf{x})p(\mathbf{z}^1 \cap \mathbf{z}^2|\mathbf{x}) + \omega p(\mathbf{z}^{2/1}|\mathbf{x})p(\mathbf{z}^1 \cap \mathbf{z}^2|\mathbf{x})} \\
& = \frac{p(\mathbf{z}^{1/2}|\mathbf{x})p(\mathbf{z}^{2/1}|\mathbf{x})}{(1-\omega)p(\mathbf{z}^{1/2}|\mathbf{x}) + \omega p(\mathbf{z}^{2/1}|\mathbf{x})} p(\mathbf{z}^1 \cap \mathbf{z}^2|\mathbf{x})p(\mathbf{x}). \tag{2.5.12}
\end{aligned}$$

□

Thus, the term  $p(\mathbf{z}^1 \cap \mathbf{z}^2|\mathbf{x})$  which accounts for the ‘rumor propagation’, is only accounted once, unlike in equation (2.3.2).

## 2.5.2 On the Entropy of Harmonic Average Density

The entropy ensures certainty of information as two densities are fused together. Though the Bayesian formula provided in equation (2.3.3) ensures optimal fusion of information, it does not contrast anything about the entropy. Though, for Gaussian distributions, it can be proved that Bayes fusion always results in a decreasing entropy, conservative fusion techniques always result in a relatively higher entropy, which explains the term ‘conservative’, since they hold back the information that could have been released if optimally fused.

The conservativeness of GMD has been proved in [8] as follows:

- Each component is replaced by a conservative density.

- Double counting of information is avoided.

Thus, the fused result is always conservative. It can be proved that the harmonic average has a higher entropy than the geometric average in the case of non-normalized densities. This can be used to develop a relationship between the entropy of harmonic and geometric average densities.

**Theorem 2.** *The HMD, in general, has a higher entropy than the GMD for the non-normalized case or when the normalization constants are equal.*

*Proof.* For  $\sum_i \omega_i = 1$ , the geometric average is given by the formula (omitting  $k$ ),

$$\mathcal{M}_\omega^g(p_1(\mathbf{x}), p_2(\mathbf{x})) = p_1(\mathbf{x})^{\omega_1} p_2(\mathbf{x})^{\omega_2} \quad (2.5.13a)$$

$$= \frac{p_1(\mathbf{x})p_2(\mathbf{x})}{p_1(\mathbf{x})^{\omega_2} p_2(\mathbf{x})^{\omega_1}}. \quad (2.5.13b)$$

Using the AM-GM property,  $AM \geq GM$ , we can write

$$\frac{p_1(\mathbf{x})p_2(\mathbf{x})}{\omega_1 p_1(\mathbf{x}) + \omega_2 p_2(\mathbf{x})} \leq \frac{p_1(\mathbf{x})p_2(\mathbf{x})}{p_1(\mathbf{x})^{\omega_1} p_2(\mathbf{x})^{\omega_2}}, \quad (2.5.14a)$$

$$\ln \frac{p_1(\mathbf{x})p_2(\mathbf{x})}{\omega_1 p_1(\mathbf{x}) + \omega_2 p_2(\mathbf{x})} \leq \ln \frac{p_1(\mathbf{x})p_2(\mathbf{x})}{p_1(\mathbf{x})^{\omega_1} p_2(\mathbf{x})^{\omega_2}}, \quad (2.5.14b)$$

where we have interchanged  $\omega_i$ 's without any loss of generality. From equations (2.5.14a) and (2.5.14b),

$$\begin{aligned} \int \frac{p_1(\mathbf{x})p_2(\mathbf{x})}{\omega_1 p_1(\mathbf{x}) + \omega_2 p_2(\mathbf{x})} \ln \frac{p_1(\mathbf{x})p_2(\mathbf{x})}{\omega_1 p_1(\mathbf{x}) + \omega_2 p_2(\mathbf{x})} d\mathbf{x} &\leq \\ \int \frac{p_1(\mathbf{x})p_2(\mathbf{x})}{p_1(\mathbf{x})^{\omega_1} p_2(\mathbf{x})^{\omega_2}} \ln \frac{p_1(\mathbf{x})p_2(\mathbf{x})}{p_1(\mathbf{x})^{\omega_1} p_2(\mathbf{x})^{\omega_2}} d\mathbf{x}. &\end{aligned} \quad (2.5.15)$$

Multiplying both sides by  $-1$  completes the proof for non-normalized case.

To establish the result between the densities, the normalization constants are taken into account. Again using equation (2.5.14a),

$$\int \frac{p_1(\mathbf{x})p_2(\mathbf{x})}{\omega_1 p_1(\mathbf{x}) + \omega_2 p_2(\mathbf{x})} d\mathbf{x} \leq \int \frac{p_1(\mathbf{x})p_2(\mathbf{x})}{p_1(\mathbf{x})^{\omega_1} p_2(\mathbf{x})^{\omega_2}}. \quad (2.5.16a)$$

$$\Rightarrow \zeta_{\mathcal{M}}^h \leq \zeta_{\mathcal{M}}^g \quad (2.5.16b)$$

This implies that

$$\mathbb{H}_h \geq \frac{\zeta_{\mathcal{M}}^g}{\zeta_{\mathcal{M}}^h} \mathbb{H}_g + \left[ \ln \zeta_{\mathcal{M}}^h - \frac{\zeta_{\mathcal{M}}^g}{\zeta_{\mathcal{M}}^h} \ln \zeta_{\mathcal{M}}^g \right]. \quad (2.5.17)$$

Note that the fraction  $\frac{\zeta_{\mathcal{M}}^g}{\zeta_{\mathcal{M}}^h}$  is greater than one. The last term on the RHS is usually negative and only vanishes when the normalization constants are equal. Due to division by a mixture density, it is non-trivial to establish a proof for normalized densities, however, for the proposed implementation method in section 2.6, the entropy of HMD is less than GMD.  $\square$

The proof that the normalization constant  $\zeta_{\mathcal{M}}^h$  is convex with respect to  $\omega$  similar to that in GMD [8] is presented below.

**Theorem 3.** *The normalization constant  $\zeta_{\mathcal{M}}^h\{p_1(\mathbf{x}_k), p_2(\mathbf{x}_k)\}$  in equation (2.5.4) it is a convex function of  $\omega$  and is less than 1 for  $0 \leq \omega \leq 1$ .*

*Proof.* Differentiating equation (2.5.4) with respect to  $\omega$  (omitting  $k$ ),

$$\begin{aligned} \frac{\partial \zeta_{\mathcal{M}}^h\{p_1(\mathbf{x}), p_2(\mathbf{x})\}}{\partial \omega} &= \frac{\partial}{\partial \omega} \int_{\mathbb{R}_n} \frac{p_1(\mathbf{x})p_2(\mathbf{x})}{(1-\omega)p_1(\mathbf{x}) + \omega p_2(\mathbf{x})} d\mathbf{x} \\ &= \int_{\mathbb{R}_n} -\frac{p_1(\mathbf{x})p_2(\mathbf{x}) [p_2(\mathbf{x}) - p_1(\mathbf{x})]}{[(1-\omega)p_1(\mathbf{x}) + \omega p_2(\mathbf{x})]^2} d\mathbf{x}. \end{aligned}$$

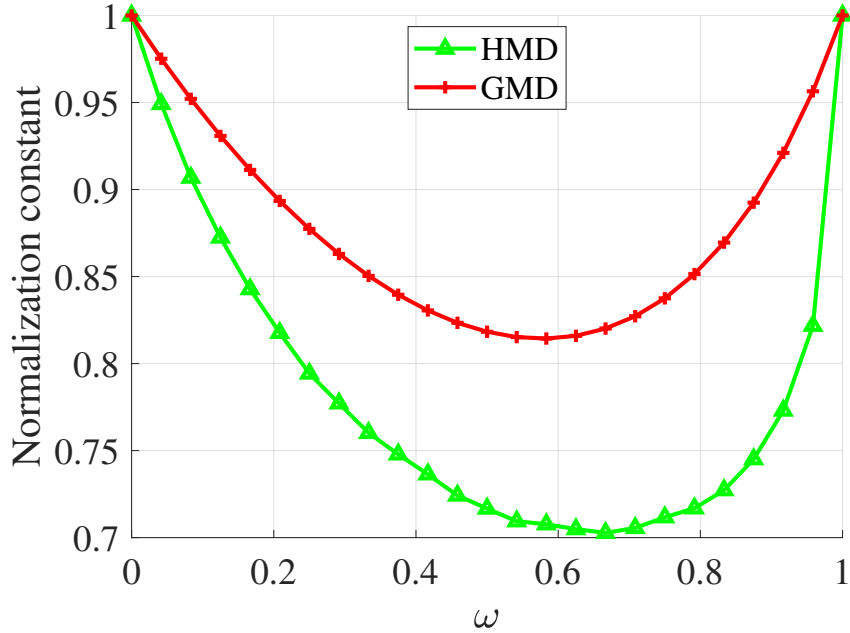


Figure 2.3: Normalization constant for various values of  $\omega$  in case of GMD (red) and HMD (green).

The second derivative is,

$$\frac{\partial^2 \zeta_{\mathcal{M}}^h \{p_1(\mathbf{x}), p_2(\mathbf{x})\}}{\partial \omega^2} = \int_{\mathbb{R}^n} 2 \frac{p_1(\mathbf{x}) p_2(\mathbf{x}) [p_2(\mathbf{x}) - p_1(\mathbf{x})]^2}{[(1 - \omega)p_1(\mathbf{x}) + \omega p_2(\mathbf{x})]^3} d\mathbf{x},$$

which is clearly positive for  $0 \leq \omega \leq 1$  since  $p_1(\mathbf{x})$  and  $p_2(\mathbf{x})$  are valid probability densities. Fig. 2.3 validates Theorem 3 by showing the comparison between  $\zeta_{\mathcal{M}}^h$  and  $\zeta_{\mathcal{M}}^g$  v/s  $\omega$  for two arbitrarily chosen densities.

It can also be noted that the bounds on  $\zeta_{\mathcal{M}}^h$  are on the limiting values of  $\omega$ ,

$$\zeta_{\mathcal{M}}^h \leq \zeta_{\mathcal{M}}^h|_{\omega=\{0,1\}} = 1. \quad (2.5.18)$$

□

### 2.5.3 Harmonic Averaging as a Recursive Fusion Strategy

For  $N$  number of sensors, the harmonic average assumes the following expression (omitting the argument in  $p_i(\mathbf{x}_k)$  for brevity),

$$\mathcal{M}_\omega^h(p_1, p_2, \dots, p_N) = \frac{\prod_i^N p_i}{\sum_i^N \omega_i \prod_{j \neq i}^N p_j}, \quad \sum \omega_i = 1. \quad (2.5.19)$$

However, instead of using the above formulation, a recursive approach to keep the complexity at the minimum can be used. Considering an example with three sensors, the harmonic average is

$$\mathcal{M}_\omega^h(p_1, p_2, p_3) = \frac{p_1 p_2 p_3}{\omega_1 p_2 p_3 + \omega_2 p_1 p_3 + \omega_3 p_1 p_2}, \quad (2.5.20)$$

$$\begin{aligned} \frac{1}{\mathcal{M}_\omega^h(p_1, p_2, p_3)} &= \frac{\omega_1}{p_1} + \frac{\omega_2}{p_2} + \frac{\omega_3}{p_3} \\ &= \frac{1}{\alpha} \left( \frac{\alpha \omega_1}{p_1} + \frac{\alpha \omega_2}{p_2} \right) + \frac{\omega_3}{p_3} \\ &= \frac{1/\alpha}{\mathcal{M}_\omega^h(p_1, p_2)} + \frac{\omega_3}{p_3}. \end{aligned} \quad (2.5.21)$$

Here  $\sum_i^3 \omega_i = 1$  and  $\alpha$  is chosen such that  $\alpha(\omega_1 + \omega_2) = 1$ . Thus, it can be shown that,

$$\mathcal{M}_\omega^h(p_1, p_2, p_3) = \mathcal{M}_{\omega, \alpha}^h[\mathcal{M}_{\omega, \alpha}^h(p_1, p_2), p_3], \quad (2.5.22)$$

which can be extended to any number of sensors. Therefore, the harmonic averaging can be performed sequentially as long as the resulting density does not diminish, ensuring the amount of certainty in the fusion process. This can also be proved as a



property of the harmonic mean density as follows.

**Theorem 4.** *The harmonic average density is bounded from below by one of the component densities.*

*Proof.* The proof is a straightforward property of the harmonic mean with respect to the components. Any abstract mean follows the property [69] (omitting  $k$ ),

$$\inf\{p_1(\mathbf{x}), p_2(\mathbf{x})\} \leq \mathcal{M}_\omega^h(p_1(\mathbf{x}), p_2(\mathbf{x})) \leq \sup\{p_1(\mathbf{x}), p_2(\mathbf{x})\}.$$

Since it has been proved in equation (2.5.18) that the normalization constant is less than 1,

$$\frac{\mathcal{M}_\omega^h(p_1(\mathbf{x}), p_2(\mathbf{x}))}{\zeta_{\mathcal{M}}^h} \geq \mathcal{M}_\omega^h(p_1(\mathbf{x}), p_2(\mathbf{x})) \geq \inf\{p_1(\mathbf{x}), p_2(\mathbf{x})\}. \quad (2.5.23)$$

Thus for the normalized pdf, the statement  $\inf\{p_1(\mathbf{x}), p_2(\mathbf{x})\} \leq \mathbf{M}_\omega^h(p_1(\mathbf{x}), p_2(\mathbf{x}))$  is valid as long as the normalization constant is less than 1, which is proved in Theorem 3. □

## 2.5.4 On the K-L Divergence of Harmonic Mean Density

**Proposition 2.5.1.** *The HMD holds the potential for being a Kullback-Leibler average.*

*Proof.* Denote by  $\mathbb{D}(p_x : p_y)$ , the Kullback-Leibler divergence between densities  $p_x$  and  $p_y$ , the aim is to prove that

$$\mathbb{D}(p_1(\mathbf{x}) : \mathbf{M}_\omega^h(p_1, p_2)) \leq \mathbb{D}(p_1(\mathbf{x}) : p_2(\mathbf{x})). \quad (2.5.24)$$

Or conversely,

$$\mathbb{D}(p_2(\mathbf{x}) : \mathbf{M}_\omega^h(p_1, p_2)) \leq \mathbb{D}(p_2(\mathbf{x}) : p_1(\mathbf{x})), \quad (2.5.25)$$

since the K-L divergence is asymmetric. First,

$$\begin{aligned} \mathbb{D}(p_1(\mathbf{x}) : \mathbf{M}_\omega^h(p_1, p_2)) &= \int_{\mathbb{R}_n} p_1(\mathbf{x}) \ln \frac{\{\omega p_1(\mathbf{x}) + (1 - \omega)p_2(\mathbf{x})\}}{p_2(\mathbf{x})} + \ln(\kappa), \\ & \end{aligned} \quad (2.5.26)$$

where  $\kappa$  denotes the normalization constant  $\zeta_{\mathcal{M}}^h$ . Denoting the cross-entropy between  $p_1(\mathbf{x})$  and  $p_2(\mathbf{x})$  by  $\mathbb{H}_{12}$ ,

$$\begin{aligned} \mathbb{D}(p_1(\mathbf{x}) : \mathbf{M}_\omega^h(p_1, p_2)) &= \int p_1(\mathbf{x}) \ln\{\omega p_1(\mathbf{x}) + (1 - \omega)p_2(\mathbf{x})\} + \mathbb{H}_{12} + \ln(\kappa) \\ &= \mathbb{D}(p_1(\mathbf{x}) : p_2(\mathbf{x})) + \int p_1(\mathbf{x}) \ln\{\omega p_1(\mathbf{x}) + (1 - \omega)p_2(\mathbf{x})\} \\ &\quad - \int p_1(\mathbf{x}) \ln p_1(\mathbf{x}) + \ln(\kappa) \\ &= \mathbb{D}(p_1(\mathbf{x}) : p_2(\mathbf{x})) - \mathbb{D}(p_1(\mathbf{x}) : p_{AA}(\mathbf{x})) + \ln(\kappa). \end{aligned} \quad (2.5.27)$$

Conversely,

$$\begin{aligned} \mathbb{D}(p_2(\mathbf{x}) : \mathbf{M}_\omega^h(p_1, p_2)) &= \mathbb{D}(p_2(\mathbf{x}) : p_1(\mathbf{x})) - \mathbb{D}(p_2(\mathbf{x}) : p_a(\mathbf{x})) + \ln(\kappa), \\ & \end{aligned} \quad (2.5.28)$$

where the standard result  $\mathbb{H}_{12} = \mathbb{D}(p_1(\mathbf{x}) : p_2(\mathbf{x})) + \mathbb{H}(p_1)$  is used and  $p_a(\mathbf{x})$  is the arithmetic mean density between  $p_1(\mathbf{x})$  and  $p_2(\mathbf{x})$ .

Note that  $\ln(\kappa)$  is less than 0 owing to equation (2.5.18) and  $\mathbb{D}(p_2(\mathbf{x}) : p_a(\mathbf{x}))$  is always positive, thus proving the proposition. Also, if  $p_1(\mathbf{x}) = p_2(\mathbf{x})$ ,  $\kappa$  turns out to be 1 from the equation as expected.  $\square$

The statement can be generalized to  $n$  number of densities  $p_i(\mathbf{x})$   $i = \{1, 2, \dots, n\}$ .

$$\mathbb{D}(p_i(\mathbf{x}) : M_\omega^h(p_1, \dots, p_n)) \leq \mathbb{D}(p_i(\mathbf{x}) : \max\{p_j(\mathbf{x})\}), \quad (2.5.29)$$

where  $j = \{1, 2, \dots, n\}$ ,  $j \neq i$ . Note that,

$$\mathbb{D}(p_i(\mathbf{x}) : p_j(\mathbf{x})) \geq \mathbb{D}(p_i(\mathbf{x}) : \max\{p_j(\mathbf{x})\}), \text{ for } j \neq i. \quad (2.5.30)$$

therefore the harmonic average density is placed in such a way that its Kullback-Leibler divergence is less than that among the least of the densities. This is analogous to the parallel resistor combination where the value of the equivalent resistance is less than or equal to that of the least resistor. In that case as well, the combination is a harmonic mean.

### 2.5.5 On the Monotonicity of Harmonic Mean Density

The shape preserving property of the harmonic average density can be proved using the monotonicity of a function of two variables. A function of two variables  $f :$

$(x, y) \in \mathbb{R}^2$  can be termed monotonic if for fixed  $(x, y)$  and  $(x', y')$

$$(x \leq x' \text{ and } y \leq y') \implies f(x, y) \leq f(x', y') \quad (2.5.31)$$

Equation. (2.5.31) can be proved for HMD as follows,

**Theorem 5.** *The harmonic mean density follows the monotonic property in equation (2.5.31).*

*Proof.* The proof here is shown for two densities  $p_1(\mathbf{x})$  and  $p_2(\mathbf{x})$  which can be extended to any number of sensors using equation (2.5.22). The gradient of the harmonic average is (omitting the variable  $\mathbf{x}$  and  $k$  for clarity).

$$\begin{aligned} \nabla(\mathbf{M}_\omega^h(p_1, p_2)) = \\ \left[ \frac{1}{\omega_1 p_1^2} \nabla(p_1) + \frac{1}{\omega_2 p_2^2} \nabla(p_2) \right] (\mathbf{M}_\omega^h(p_1, p_2))^2. \end{aligned} \quad (2.5.32)$$

It can be seen that for fixed  $p_1(\mathbf{x})$ , the harmonic average is monotonic with respect to  $p_2(\mathbf{x})$  and monotonic w.r.t.  $p_1(\mathbf{x})$  when the other is fixed. Thus from equation (2.5.32),

$$\text{if } \tilde{p}_1 \leq p_1 \implies \mathbf{M}_\omega^h(\tilde{p}_1, \gamma) \leq \mathbf{M}_\omega^h(p_1, \gamma), \quad (2.5.33a)$$

and,

$$\text{if } \tilde{p}_2 \leq p_2 \implies \mathbf{M}_\omega^h(\gamma, \tilde{p}_2) \leq \mathbf{M}_\omega^h(\gamma, p_2), \quad (2.5.33b)$$

where  $\gamma$  is an arbitrary constant. The proof of equation (2.5.31) for harmonic average

is a direct consequence of equations (2.5.33a) and (2.5.33b).  $\square$

## 2.6 Implementation

The major issue in implementation of HMD is its denominator which takes the form of a valid probability density mixture. Closed form solutions for division by a mixture do not exist [3] hence approximate solutions are used.

Considering Gaussian densities, the simplest solution to the division problem is to use a normal approximation for the denominator,

$$\omega \mathcal{N}(\hat{\mathbf{x}}_k^1, \mathbf{\Gamma}_k^1) + (1 - \omega) \mathcal{N}(\hat{\mathbf{x}}_k^2, \mathbf{\Gamma}_k^2) \approx \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^{eq}, \mathbf{\Gamma}_k^{eq}), \quad (2.6.1)$$

where  $\hat{\mathbf{x}}_k^{eq}$  and  $\mathbf{\Gamma}_k^{eq}$  are respectively the mean and covariance of the resulting Gaussian approximation of the mixture,

$$\hat{\mathbf{x}}_k^{eq} \triangleq \sum_{m=1}^M \mu^m \hat{\mathbf{x}}_k^m, \quad (2.6.2)$$

$$\mathbf{\Gamma}_k^{eq} \triangleq \sum_{m=1}^M \mu^m [\mathbf{\Gamma}_k^m + (\hat{\mathbf{x}}_k^{eq} - \hat{\mathbf{x}}_k^m)(\hat{\mathbf{x}}_k^{eq} - \hat{\mathbf{x}}_k^m)^T]. \quad (2.6.3)$$

Let  $p(\mathbf{x}_k | z_k^i) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^i, \mathbf{\Gamma}_k^i)$  and  $p(\mathbf{x}_k | z_k^j) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^j, \mathbf{\Gamma}_k^j)$ , then using the Gaussian equivalent for denominator,

$$p(\mathbf{x}_k | \mathbf{z}_k^i \cup \mathbf{z}_k^j) = \frac{\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^i, \mathbf{\Gamma}_k^i) \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^j, \mathbf{\Gamma}_k^j)}{\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^{eq}, \mathbf{\Gamma}_k^{eq})} \quad (2.6.4)$$

$$\propto \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^f, \mathbf{\Gamma}_k^f), \quad (2.6.5)$$

where the superscript  $f$  stands for *fused*. The fused mean and covariance are then,

$$\mathbf{\Gamma}_k^f = \left( \mathbf{\Gamma}_k^{i^{-1}} + \mathbf{\Gamma}_k^{j^{-1}} - \mathbf{\Gamma}_k^{eq^{-1}} \right)^{-1}, \quad (2.6.6a)$$

$$\hat{\mathbf{x}}_k^f = \mathbf{\Gamma}_k^f \left[ \left( \mathbf{\Gamma}_k^{i^{-1}} \hat{\mathbf{x}}_k^i + \mathbf{\Gamma}_k^{j^{-1}} \hat{\mathbf{x}}_k^j \right) - \mathbf{\Gamma}_k^{eq^{-1}} \hat{\mathbf{x}}_k^{eq} \right], \quad (2.6.6b)$$

where standard results on division and product of Gaussian densities have been used [2, 14].  $\hat{\mathbf{x}}_k^{eq}$  and  $\mathbf{\Gamma}_k^{eq}$  are as mentioned in equations (2.6.2) and (2.6.3) respectively for weights  $\omega$  and  $(1 - \omega)$ .

Note that the division of the Gaussian distributions is only valid for certain conditions which is met by HMD and can be proved as follows.

**Proposition 2.6.1.** *The division of Gaussian densities as performed in equation (2.6.5) is always valid.*

*Proof.* See appendix A. □

Note that the Gaussian approximation of the mixture only changes our estimate of the common information which itself was not exact. Thus it does not affect the quality of fusion in general. Hence,

$$\begin{aligned} p(\mathbf{x}_k | \mathbf{z}_k^1 \cap \mathbf{z}_k^2) &\approx \omega_1(p(\mathbf{x}_k | \mathbf{z}_k^1) + \omega_2(p(\mathbf{x}_k | \mathbf{z}_k^2) \\ &\approx \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^{eq}, \mathbf{\Gamma}_k^{eq}). \end{aligned} \quad (2.6.7)$$

It is worth noting that the proposed method of implementation of HMD is similar to different extensions of CI with corresponding interpretations for the common information component  $(\hat{\mathbf{\Gamma}}, \mathbf{\Gamma})$ . For instance, the ellipsoidal intersection (EI) provides

the fused result at time  $k$  as,

$$\mathbf{\Gamma}_k^f = \left[ \mathbf{\Gamma}_k^{i-1} + \mathbf{\Gamma}_k^{j-1} - \mathbf{\Gamma}_k^{EI-1} \right]^{-1} \quad (2.6.8)$$

$$\hat{\mathbf{x}}_k^f = \mathbf{\Gamma}_k^f \left[ \mathbf{\Gamma}_k^{i-1} \hat{\mathbf{x}}_k^i + \mathbf{\Gamma}_k^{j-1} \hat{\mathbf{x}}_k^j - \hat{\mathbf{\Gamma}}_k^{EI} \mathbf{\Gamma}_k^{EI-1} \right] \quad (2.6.9)$$

See [85] for the interpretations of  $(\hat{\mathbf{\Gamma}}_k^{EI}, \mathbf{\Gamma}_k^{EI})$ . Similarly, the inverse covariance intersection (ICI) in [73] suggests that the common information is a weighted arithmetic average of the components.

$$\mathbf{\Gamma}_k^f = \left[ \mathbf{\Gamma}_k^{i-1} + \mathbf{\Gamma}_k^{j-1} - \mathbf{\Gamma}_k^{ICI-1} \right]^{-1} \quad (2.6.10)$$

$$\hat{\mathbf{x}}_k^f = \mathbf{\Gamma}_k^f \left[ \mathbf{\Gamma}_k^{i-1} \hat{\mathbf{x}}_k^i + \mathbf{\Gamma}_k^{j-1} \hat{\mathbf{x}}_k^j - \left( \omega \mathbf{\Gamma}_k^{ICI-1} \hat{\mathbf{x}}_k^i + (1 - \omega) \mathbf{\Gamma}_k^{ICI-1} \hat{\mathbf{x}}_k^j \right) \right] \quad (2.6.11)$$

where,

$$\mathbf{\Gamma}_k^{ICI} = \omega \mathbf{\Gamma}_k^i + (1 - \omega) \mathbf{\Gamma}_k^j \quad (2.6.12)$$

The structure of common information in this case is very similar to the proposed method except the spread of means term in the Gaussian mixture equivalent of Eqn. (57). The difference is due to the fact that HMD uses arithmetic average of local densities rather than that of parameters as in the case of ICI. When fusing estimates with equal means, the proposed method matches exactly with ICI. The interpretation of spread-of-means is discussed elaborately in subsequent section.

The scope of this work is restricted to the use of the Gaussian approximation of the common information, excluding the exact development as a scope for future work.

**Implications of the spread of means term in eqn. (2.6.3)** The reader might believe that the proposed method of implementation is dubious in the sense that one can increase the spread of means term by arbitrarily increasing the Euclidean distance between individual means. However, this is not true if the densities are tested for association before fusion. The association test in such a case would reject the hypothesis that local densities belong to the same target if the normalized distance between the means is large. For a positive association test, the squared distance between the means has to be less than (scalar case),

$$(\mu_1 - \mu_2)^2 \leq \gamma_\alpha \times (\sigma_1^2 + \sigma_2^2 - 2\sigma_{12}) \quad (2.6.13)$$

where  $\gamma$  is a threshold, with  $(1 - \alpha)$  as the confidence level.  $\mu_i$  and  $\sigma_i$  are the  $i^{\text{th}}$  mean and standard deviation, respectively.  $\sigma_{12} = \rho\sqrt{\sigma_1\sigma_2}$  is the correlation coefficient between the two estimates, with  $\rho$  as a linear correlation coefficient. Thus, arbitrarily large separations between the individual means are rejected by the test.

Another implication is that if the local densities are assumed to be consistent, their estimate should correspond to their respective mean square error (MSE). Since both the estimates represent the same quantity, they should statistically lie within their respective covariance ellipsoids. Thus, local consistency is required for the effective implementation of HMD.

It should also be noted that the proposed implementation might result in lower covariance than the local optimum see [12, see eqn. (8.4.4-5)]. However, as per the definition of consistency in [8], the proposed implementation is inconsistent only if one of the densities is inconsistent.

It can be seen that using the same formulation allows us to deal with the scenario



when track densities are a Gaussian mixture

$$p(\mathbf{x}_k | \mathbf{z}_k^1) = \sum_i^M \alpha_i \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^i, \mathbf{\Gamma}_k^i),$$

$$p(\mathbf{x}_k | \mathbf{z}_k^2) = \sum_j^N \beta_j \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^j, \mathbf{\Gamma}_k^j).$$

Then, the fused density is another Gaussian mixture with the  $M \times N$  number of modes and,

$$p(\mathbf{x}_k | \mathbf{z}_k^1 \cup \mathbf{z}_k^2) = \sum_i^M \sum_j^N \alpha_i \beta_j \kappa_{ij} \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^f, \mathbf{\Gamma}_k^f), \quad (2.6.14)$$

where  $\hat{\mathbf{x}}_k^f$  and  $\mathbf{\Gamma}_k^f$  are exactly the same as in equations (2.6.6b) and (2.6.6a). The scaling factor  $\kappa_{ij}$  is the result of product of Gaussian densities and division by a Gaussian mixture equivalent. It is given by,

$$\kappa_{ij} = \frac{\mathcal{N}(\hat{\mathbf{x}}_k^j; \hat{\mathbf{x}}_k^i, \mathbf{\Gamma}_k^i + \mathbf{\Gamma}_k^j)}{\mathcal{N}(\hat{\mathbf{x}}_k^m; \hat{\mathbf{x}}_k^{\text{naive},ij}, \mathbf{\Gamma}_k^m - \mathbf{\Gamma}_k^{\text{naive},ij})} \times \frac{|\mathbf{\Gamma}_k^m - \mathbf{\Gamma}_k^{\text{naive},ij}|}{|\mathbf{\Gamma}_k^{\text{naive},ij}|} \quad (2.6.15)$$

where,  $(\hat{\mathbf{x}}_k^{\text{naive},ij}, \mathbf{\Gamma}_k^{\text{naive},ij})$  is the result of naive fusion between  $i^{\text{th}}$  and  $j^{\text{th}}$  densities. The quantities  $(\hat{\mathbf{x}}_k^m, \mathbf{\Gamma}_k^m)$  arise from the Gaussian approximation of the denominator

mixture,

$$\begin{aligned} \omega_1 p(\mathbf{x}_k | \mathbf{z}_k^1) + \omega_2 p(\mathbf{x}_k | \mathbf{z}_k^2) &= \omega_1 \sum_i^M \alpha_i \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^i, \mathbf{\Gamma}_k^i) + \omega_2 \sum_j^N \beta_j \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^j, \mathbf{\Gamma}_k^j), \\ &= \sum_m^{M+N} \mu^m \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^m, \mathbf{\Gamma}_k^m), \quad m = \begin{cases} i, & 1 \leq m \leq M \\ j, & M+1 \leq m \leq M+N \end{cases} \end{aligned} \quad (2.6.16)$$

where,

$$\mu^m = \begin{cases} \omega_1 \alpha_i, & 1 \leq m \leq M \\ \omega_2 \beta_j, & M+1 \leq m \leq M+N. \end{cases}$$

### 2.6.1 On the evaluation of Fusion weights

A missing aspect of the conservative fusion strategies in the literature is the evaluation of fusion weights  $\omega_i$ . For instance, in the geometric averaging, the common information which is also the denominator in the exact Bayesian formulation of equation (2.3.3) was approximated to be (see equation (2.5.9)),

$$p(\mathbf{x}_k | \mathbf{z}_k^1 \cap \mathbf{z}_k^2) = p(\mathbf{x}_k | \mathbf{z}_k^1)^{1-\omega} p(\mathbf{x}_k | \mathbf{z}_k^2)^\omega. \quad (2.6.17)$$

The quality of any conservative fusion strategy in the Bayesian sense lies in the statement - *“How close is the common information to its approximation”*. There is no substantial research in finding the optimal fusion weights except in [98]. An alternate direction based on common information is provided. Using the concept of conditional independence depicted in Fig. 2.5 and the Bayes’ theorem, the conditional

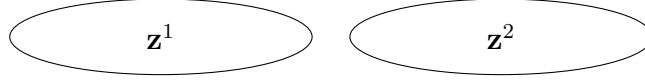


Figure 2.4: Independent Dataset

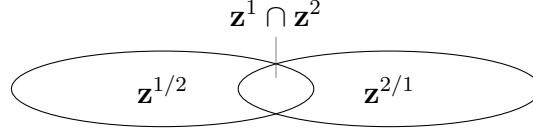


Figure 2.5: Non-independent Dataset [48]

density based on common information can be written as,

$$\begin{aligned} P(\mathbf{z}^i|\mathbf{x}) &= P(\mathbf{z}^{i/j}|\mathbf{x})P(\mathbf{z}^i \cap \mathbf{z}^j|\mathbf{x}) \\ \Rightarrow P(\mathbf{x}|\mathbf{z}^i \cap \mathbf{z}^j) &= P(\mathbf{x}|\mathbf{z}^i) \frac{P(\mathbf{z}^i|\mathbf{x})}{P(\mathbf{z}^i \cap \mathbf{z}^j)} \propto \frac{P(\mathbf{x}|\mathbf{z}^i)}{P(\mathbf{x}|\mathbf{z}^{i/j})}, \end{aligned} \quad (2.6.18)$$

and using the fact that  $P(\mathbf{x}|\mathbf{z}^i \cap \mathbf{z}^j) = P(\mathbf{x}|\mathbf{z}^j \cap \mathbf{z}^i)$ ,

$$P(\mathbf{x}|\mathbf{z}^i \cap \mathbf{z}^j) \propto 0.5 \frac{P(\mathbf{x}|\mathbf{z}^i)}{P(\mathbf{x}|\mathbf{z}^{i/j})} + 0.5 \frac{P(\mathbf{x}|\mathbf{z}^j)}{P(\mathbf{x}|\mathbf{z}^{j/i})}. \quad (2.6.19)$$

The above statement is appropriate since it can be seen that if independence is assumed, then  $P(\mathbf{x}|\mathbf{z}^i) = P(\mathbf{x}|\mathbf{z}^{i/j})$  and  $P(\mathbf{x}|\mathbf{z}^i \cap \mathbf{z}^j)$  comes out to be a constant. The expression for the common information in harmonic mean density is the arithmetic average of the posterior densities. Thus the fusion weights should follow

$$\omega_1 p(\mathbf{x}|\mathbf{z}^i) + \omega_2 p(\mathbf{x}|\mathbf{z}^j) \approx 0.5 \frac{P(\mathbf{x}|\mathbf{z}^i)}{P(\mathbf{x}|\mathbf{z}^{i/j})} + 0.5 \frac{P(\mathbf{x}|\mathbf{z}^j)}{P(\mathbf{x}|\mathbf{z}^{j/i})}. \quad (2.6.20)$$

The term  $P(\mathbf{x}|\mathbf{z}^{i/j})$  is referred to as the *tracklet* in the literature and has been extensively researched [37, 38]. The arithmetic average on the L.H.S of equation (2.6.20)

which is a mixture density should approximate the mixture on the R.H.S as close as possible. The simple approximation of the common information is an important advantage of using the harmonic fusion which could be solved using EM algorithm. However, there is no attempt to obtain a solution to equation (2.6.20) in this work but is left as a candidate for future research.

### 2.6.2 An Empirical Study

The properties of HMD elucidated in the previous section are observed and compared with the other conservative fusion strategies. Two correlated Gaussian densities  $p_1(\mathbf{x})$  and  $p_2(\mathbf{x})$  with mean  $\hat{\mathbf{x}}_1 = [1, 3]^T$  and  $\hat{\mathbf{x}}_2 = [7, 10]^T$ , and covariance  $\Gamma_1 = 100\mathbf{I}_2$  and  $\Gamma_2 = 50\mathbf{I}_2$  are fused as per the GMD in equation (2.4.6), AMD in equation (2.4.12) as well as the proposed fusion strategy. The dependence is quantified as,

$$\mathbf{\Gamma}_{12} = \rho\sqrt{\mathbf{\Gamma}_1}\sqrt{\mathbf{\Gamma}_2}, \quad (2.6.21)$$

where  $\rho$  is the correlation coefficient, taking only positive values between 0 and 1 (tracking applications with homogeneous sensors rarely possess negative correlation for the same target). The optimal density is calculated using the maximum likelihood approach, taking into account the correlation matrix  $\mathbf{\Gamma}_{12}$ . The entropy of various fused densities is plotted with respect to values of the fusion weight  $\omega$  in Fig. 2.6. It can be seen that the harmonic mean density possesses the highest entropy, proving its conservativeness. The arithmetic mean density always results in a Gaussian mixture, so a Gaussian approximation is taken in this example, due to which the entropy presented in Fig. 2.6 is a mere upper bound. The value of the fusion weight used here is 0.5. The resulting fused density functions are plotted in Fig. 2.7. It can

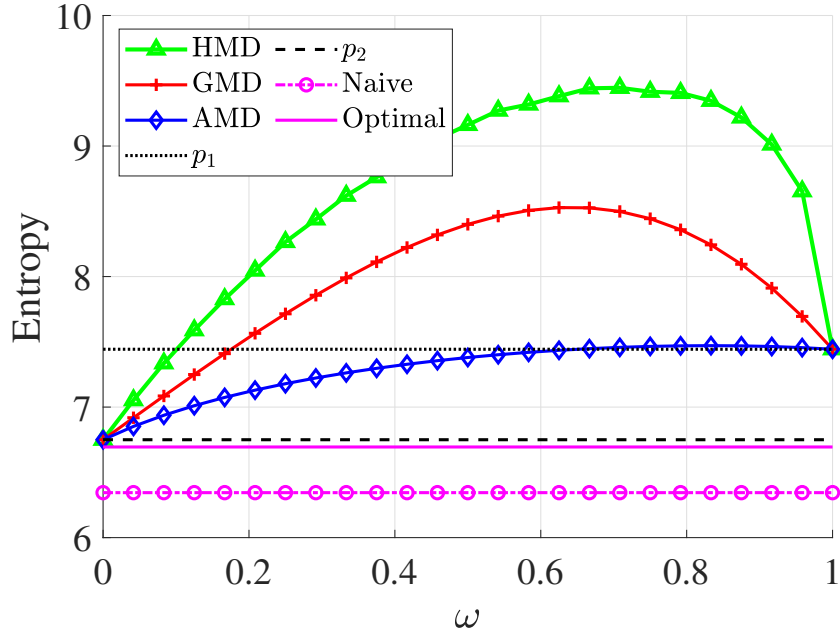


Figure 2.6: Entropy of the fused density using various methods for a correlation coefficient of  $\rho = 0.5$  among local densities.

be observed that both HMD and GMD are close to the optimal density whereas the arithmetic mean has the largest variance. For different values of  $\rho$ , the plots of conservative densities will remain the same if  $\omega$  is unchanged.

To evaluate the run time of fusion strategies, the dimensions of the Gaussian densities along with the number of components for the Gaussian mixtures are increased. The plots for computation time over 500 Monte-Carlo runs are shown in Fig. 2.8. It can be seen in Fig. 2.8a that for the Gaussian case, HMD is slightly slower. The GMD and Naive fusion require almost equal computation since both need two inverse operations whereas HMD requires three (for fusing two densities). It is also observed that the AMD has a higher average run time in the case of Gaussian density, which is due to the fact that the overhead of computing the Gaussian approximation of the resulting mixture has also been added. Also, note that in the case of Gaussian

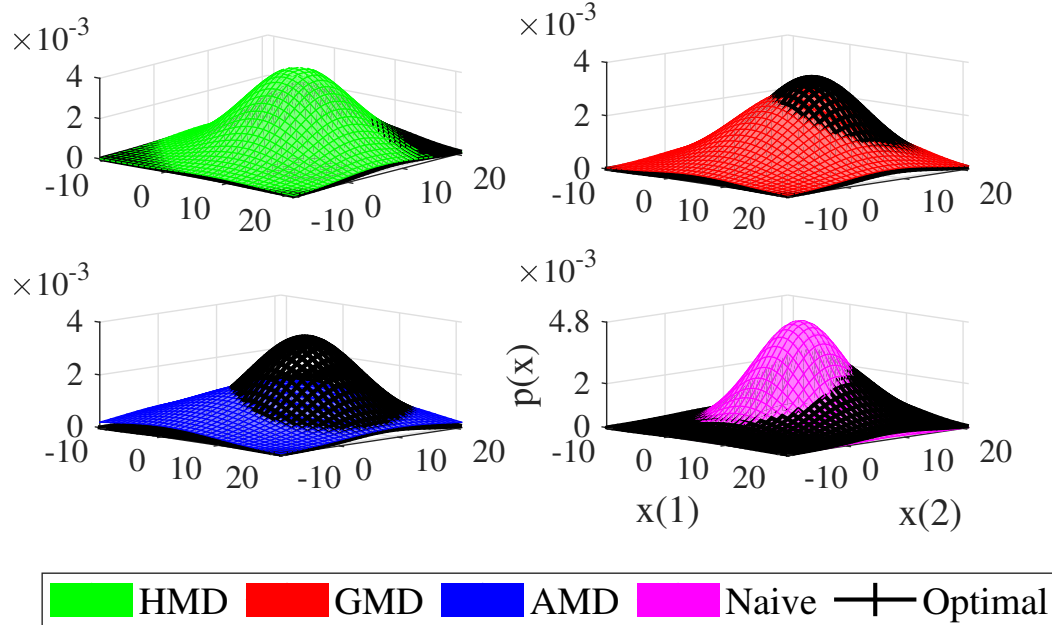


Figure 2.7: Fused density using various methods for a correlation coefficient of  $\rho = 0.5$  among local densities.

mixtures in Fig. 2.8b, pseudo-Chernoff fusion (PCF) is employed, which is the fastest method for fusing Gaussian mixtures using GMD [49] and poses inaccuracies when fusing densities are spatially closer to each other. Also, [43] states that the sigma-point approximation-based method is six times more computationally costly than PCF. In either case, HMD is as computationally efficient as GMD and naive fusion, respectively, without the requirement of any additional changes to the framework. The results were obtained on MATLAB<sup>®</sup> 2020 on a computer with Intel<sup>®</sup> Core<sup>™</sup> i7-9750H CPU @ 2.6 GHz and 16 GB of RAM.

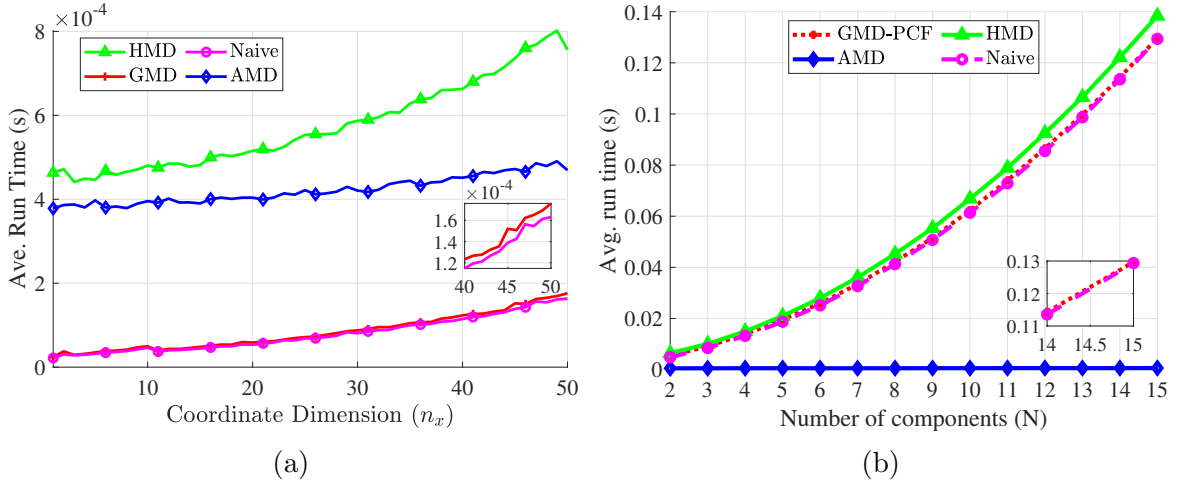


Figure 2.8: (a) Average run time v/s dimension of the Gaussian density and, (b) average run time v/s number of components in a Gaussian mixture.

## 2.7 Simulation

To test the performance of the proposed strategy in a real-time target tracking environment, two simulation scenarios are presented, one of which is a near constant velocity (NCV) target in 3D, comprising three sensors and the other is a 2D maneuvering target, tracked using two bearing-only sensors using IMM. Thus, the scenarios cover the fusion of both uni-modal and multi-modal Gaussian densities and are elaborated in different subsections described below. Note that no effort has been made to evaluate the optimal value of the fusion weight  $\omega$  and a typical value of 0.5 is taken. The baseline methods considered for the comparison are :

- Geometric Mean Density (GMD): Covariance intersection formulation for Gaussian densities.
- Arithmetic Mean Density (AMD): Weighted amalgamation of local track densities without any pruning or merging.

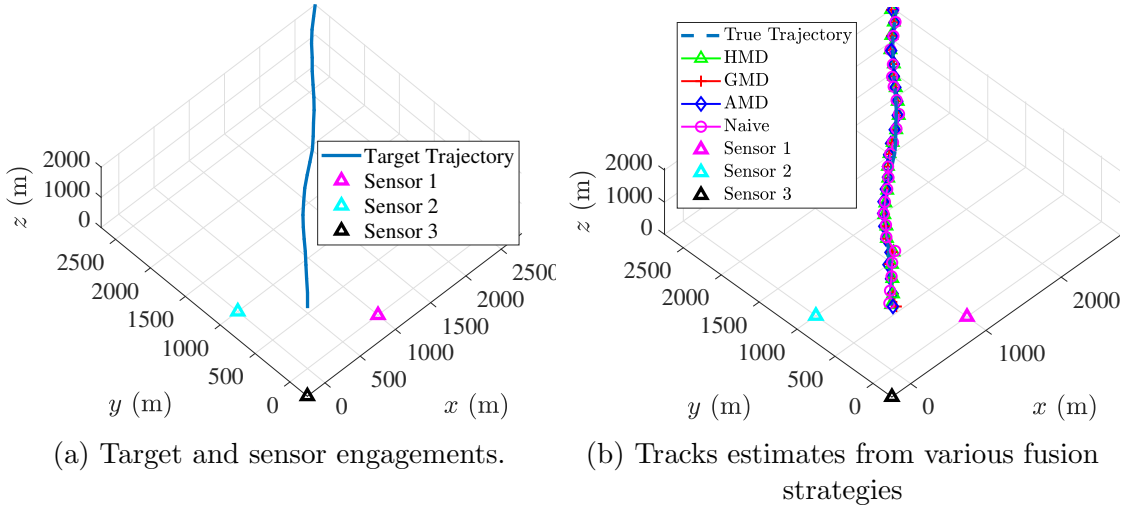


Figure 2.9: Scenario 1

- Centralized EKF: Centralized fusion of measurements using the extended Kalman filter (EKF) used as a lower bound.

### 2.7.1 Three Sensor NCV Scenario

In this case, a 3D target trajectory is generated using the near constant velocity motion model with a high process noise intensity, as shown in Fig. 2.9a, where the target starts at the origin at an altitude of 2000 meters. The state vector  $\mathbf{x}_k \in \mathbb{R}^6$  comprises of 3D Cartesian position and velocities, i.e.,  $\mathbf{x}_k = \begin{bmatrix} x_k & \dot{x}_k & y_k & \dot{y}_k & z_k & \dot{z}_k \end{bmatrix}$ . The local nodes employ an extended Kalman filter (EKF) with no feedback from the fusion center. Thus, the tracks to be fused are Gaussian densities.

The discrete time target motion and sensors measurements are modeled by the



following equations

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{w}_k, \quad (2.7.1a)$$

$$\mathbf{z}_k = \mathbf{h}(x_k, y_k, z_k) + \mathbf{v}_k, \quad (2.7.1b)$$

where,

$$\mathbf{F} = \begin{bmatrix} \mathbf{I}_3 & \Delta T \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \quad \text{and, } \mathbf{h}(x_k, y_k, z_k) = \begin{bmatrix} r_k & \theta_k & \phi_k \end{bmatrix}^T,$$

where,

$$r_k = \sqrt{x_k^2 + y_k^2 + z_k^2}; \quad \theta_k = \tan^{-1} \left( \frac{y_k}{x_k} \right); \quad \text{and, } \phi_k = \tan^{-1} \left( \frac{z_k}{\sqrt{x_k^2 + y_k^2}} \right).$$

$\Delta T$  is the sampling time, and  $\mathbf{w}_k$ ,  $\mathbf{v}_k$  are uncorrelated, zero-mean Gaussian distributed process noise vector and measurement noise vector respectively, such that  $\forall \{k, j\}$ ,

$$E[\mathbf{w}_k \mathbf{w}_j] = \delta_{kj} \mathbf{Q}; \quad E[\mathbf{w}_k \mathbf{v}_j] = \mathbf{0}; \quad E[\mathbf{v}_k \mathbf{v}_j] = \delta_{kj} \mathbf{R},$$

where  $\mathbf{Q}$  and  $\mathbf{R}$  are the process noise and measurement noise covariance matrix respectively,

$$\mathbf{Q} = \tilde{q} \begin{bmatrix} \frac{\Delta T^3}{3} \mathbf{I}_3 & \frac{\Delta T^2}{2} \mathbf{I}_3 \\ \frac{\Delta T^2}{2} \mathbf{I}_3 & \Delta T \mathbf{I}_3 \end{bmatrix}, \quad \mathbf{R} = \text{diag}(\sigma_r^2, \sigma_\theta^2, \sigma_\phi^2),$$

and  $\tilde{q}$  is the process noise intensity in meter<sup>2</sup>/second<sup>3</sup>. The parameters of the scenario

are stated in Table 2.1. Note that separate values for process noise intensity in each coordinate have been included.

Table 2.1: Parameters : Scenario 1

Parameter	Value
$\Delta T$	2 seconds.
$\sigma_r^1, \sigma_\theta^1, \sigma_\phi^1$	100 m, $2^\circ$ , $2^\circ$
$\sigma_r^2, \sigma_\theta^2, \sigma_\phi^2$	100 m, $2^\circ$ , $2^\circ$
$\sigma_r^3, \sigma_\theta^3, \sigma_\phi^3$	100 m, $1.5^\circ$ , $1.5^\circ$
$\tilde{q}_x$	$0.5 \text{ m}^2/\text{sec}^3$
$\tilde{q}_y$	$0.5 \text{ m}^2/\text{sec}^3$
$\tilde{q}_z$	$0.001 \text{ m}^2/\text{sec}^3$
Simulation time	120 seconds.

The local trackers transmit their estimates every  $2\Delta T$  to the fusion center, which are then fused together using the existing conservative fusion strategies discussed in the previous sections. The proposed HMD is compared with the AMD and the GMD (covariance intersection), where the centralized version of EKF has been chosen as the lower bound. The performance is evaluated on the basis of root-mean-square errors (RMSE) of position and velocities over 500 Monte-Carlo (MC) runs. The consistency of the fusion approaches has also been compared with the help of normalized estimation error squared (NEES) which is calculated as

$$\text{NEES}_k = \frac{1}{M} \sum_{m=1}^M (\hat{\mathbf{x}}_{k|k}^m - \mathbf{x}_k)^T \mathbf{\Gamma}^{m-1} (\hat{\mathbf{x}}_{k|k}^m - \mathbf{x}_k). \quad (2.7.2)$$

The fused scenario with various tracks are shown in Fig. 2.9b.

The position RMSE plot is shown in Fig. 2.10a, where the centralized EKF performs best as expected. It can be seen that the proposed harmonic mean density performs marginally better as compared to the other conservative density approaches.

The proposed HMD error plot converges faster than other approaches and suggests an improvement of roughly double when compared to the naive density, which starts to diverge at the end of simulation due to rumor propagation. The computation time for Gaussian density fusion for HMD in Fig. 2.8a is also almost equal to that of AMD and 50% higher than the covariance intersection.

A similar trend can be seen in the velocity RMSE plot in Fig. 2.10b with the centralized approach performing best, followed by the proposed HMD, GMD, AMD, and then the naive fusion. The velocity estimate in the case of HMD states an improvement of roughly 50% over naive fusion in terms of accuracy.

We present the average NEES plotted over 500 MC runs in Fig. 2.11 to show the consistency of the fused estimates. The 95% confidence interval for 6-dimensional state vector is shown by a blue line in the plot. It can be seen that no point in the naive plot lies inside the confidence region, proving its inconsistency. Ignoring the initial transient region, only one point out of 120 time steps lies outside the region in the case of HMD, which is acceptable. The other approaches are well below the threshold, especially the arithmetic mean density, which exhibits its over-conservativeness.

### 2.7.2 Two Sensor Passive Maneuvering Target Tracking

A major advantage of using HMD is the fusion of multi-modal and uni-modal Gaussian densities without using additional machinery, like in GMD, where a separate mechanism is required to compute accurate power of a Gaussian mixture, or AMD where variance deflation techniques are necessary to prevent the estimates from being over-conservative. To illustrate this advantage, a two-sensor multiple model based

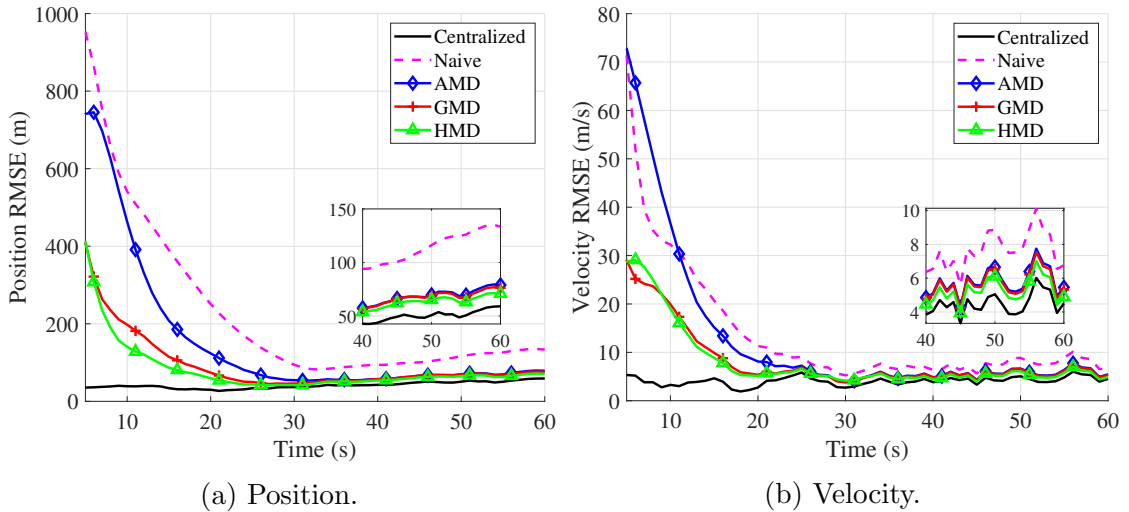


Figure 2.10: Root mean square error (RMSE) performance : Scenario 1.

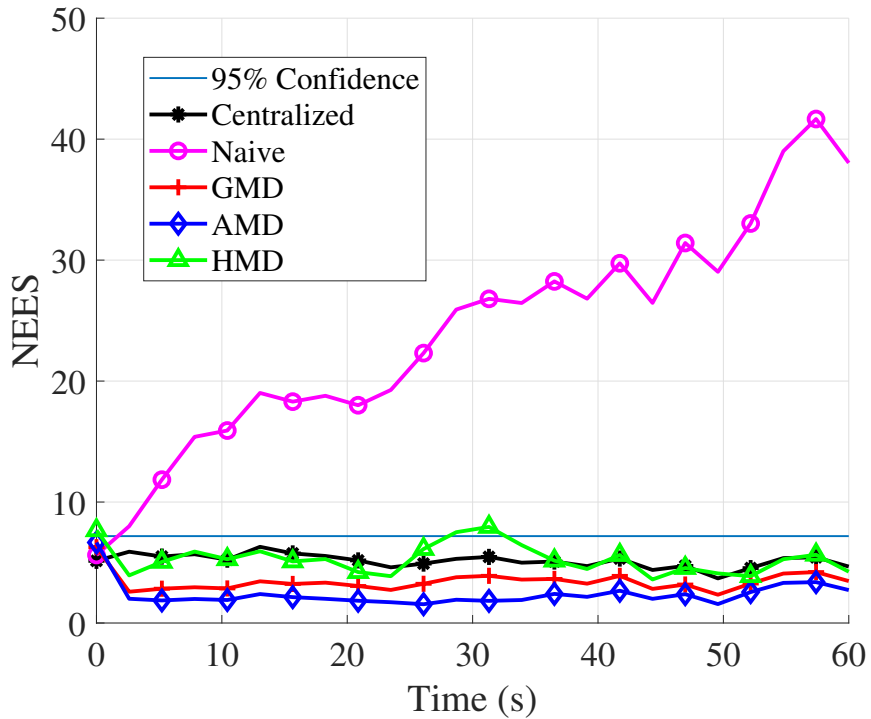


Figure 2.11: Normalized estimation error squared (NEES) for scenario 1. The HMD can be seen to be fairly consistent along with other strategies.

maneuvering target tracking using bearing-only sensor in 2D was designed. Each local tracker is an IMM consisting of a Gaussian noise NCV model (as discussed in the previous subsection) and a Wiener noise nearly constant acceleration (NCA) model. This scenario is considered non-trivial because of the following reasons :

- Unobservability of local trackers.
- Different state dimensions of the two models used in the local IMM tracker (NCA versus NCV).
- Feedback requirement from the fusion center to improve local estimates.
- Motion model mismatch between the actual target and the one employed by local trackers.

To solve the problem of growing fused mixture size, a pruning operation is performed after fusion so that the final mixture consists of two modes. This mixture is sent to the local nodes after each fusion instant as feedback (see Fig. 2.12). For instance, the GMD fusion of two bi-modal Gaussian mixtures outputs a fused density with four modes. Since the local IMM trackers employ bi-modal densities, the fused mixture size has to be pruned before transmitting it back. The different dimensions of the two modes are taken care of by padding extra zeros in the estimate of the NCV model. This poses problems while using sigma-point Chernoff fusion [43] where Cholesky decomposition needs to be carried out, thus, PCF from equation (2.4.7) [49] was used instead, which is also the fastest method.

The scenario is shown in Fig. 2.13a. The target moves in a wavelike motion about a straight line. Such trajectories are common in underwater warfare engagement

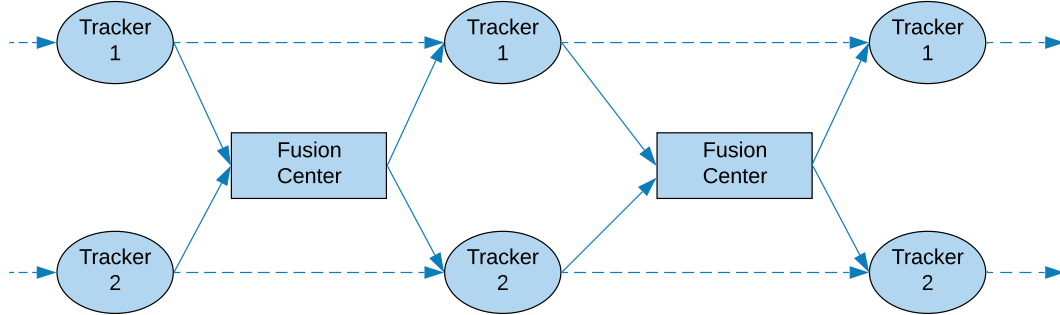


Figure 2.12: Information flow between fusion center and local trackers for scenario 2.

scenarios and buoyancy-driven underwater gliders [51]. In this case, the target was generated by rotating a sine wave motion in  $x$  and  $y$  coordinates at a fixed angle. No noise has been added to the true target trajectory due to the presence of high maneuvers already. An attempt to track this target using conservative fusion strategies with IMM was made. In addition to the baseline methods mentioned in the previous sections, the centralized methods – centralized-CV and centralized-CA were added to check if non-IMM centralized approaches are suited for the problem.

The motion model used in the IMM trackers remains the same as mentioned in equation (2.7.1) modified to 2D, with the addition of a separate NCA model categorized by the following parameters,

$$\mathbf{F}_{NCA} = \begin{bmatrix} \mathbf{I}_2 & \Delta T \mathbf{I}_2 & \frac{\Delta T^2}{2} \mathbf{I}_2 \\ \mathbf{0}_2 & \mathbf{I}_2 & \Delta T \mathbf{I}_2 \\ \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{I}_2 \end{bmatrix}, \quad \mathbf{Q}_{NCA} = \tilde{q}_{NCA} \begin{bmatrix} \frac{\Delta T^5}{20} \mathbf{I}_2 & \frac{\Delta T^4}{8} \mathbf{I}_2 & \frac{\Delta T^3}{6} \mathbf{I}_2 \\ \frac{\Delta T^4}{8} \mathbf{I}_2 & \frac{\Delta T^3}{3} \mathbf{I}_2 & \frac{\Delta T^2}{2} \mathbf{I}_2 \\ \frac{\Delta T^3}{6} \mathbf{I}_2 & \frac{\Delta T^2}{2} \mathbf{I}_2 & \Delta T \mathbf{I}_2 \end{bmatrix}. \quad (2.7.3a)$$

The sensor measurements are now bearings measured from the true north, thus

$$\theta_k = \mathbf{h}(x_k, y_k) + \mathbf{v}_k, \quad (2.7.4a)$$

where,

$$\mathbf{h}(x_k, y_k) = \tan^{-1} \left( \frac{x_k}{y_k} \right). \quad (2.7.4b)$$

The parameters used in the models are tabulated in Table 2.2. It is to be noted that the target in Fig. 2.13a does not adhere to either the NCA or NCV motion models; the plotted trajectory is completely independent of the motion statistic used, as is the case in real-time target tracking. Therefore, there is a high certainty of errors due to model mismatch. The sensors are located 600 meters apart, the target starts at [150m, 150m] and moves north-east with a velocity of 16 knots.

The results over 500 Monte-Carlo runs are plotted in Figs. [2.14a – 2.13b]. As expected, the centralized EKF employing the NCA model has the least RMSE, followed by the proposed HMD. An interesting point to note is that the centralized EKF-CV performs poorly in terms of RMSE when compared to HMD and GMD, whereas AMD failed to track the target owing to extremely high track-loss<sup>1</sup> (> 90%) while no track-loss was observed in other techniques.

---

<sup>1</sup>Track-loss is defined as the divergence of the estimate and is quantified when the final position error  $e_f$  exceeds a set threshold  $\tau$  (500m in this case),

$$e_f = \sqrt{(x_f - \hat{x}_f)^2 + (y_f - \hat{y}_f)^2} \geq \tau,$$

where  $f$  is final time index. The number of track-loss is usually represented as % of Monte-Carlo runs. Note that while calculating the RMSE, the diverged tracks are removed

### Failure of AMD in unobservable systems

The AMD is an amalgamation that only works when the estimates to be fused are sufficiently close. But since the local nodes in this scenario are non-observable, the *closeness* is not guaranteed. For instance, suppose the two estimates to be fused are  $x_1 \sim \mathcal{N}(50, 10)$  and  $x_2 \sim \mathcal{N}(-30, 20)$ . The resulting densities are  $\mathcal{N}(10, 1615)$ ,  $\mathcal{N}(23.39, 6.69)$ ,  $\mathcal{N}(23.33, 13.33)$  and  $\mathcal{N}(23.33, 6.67)$  for AMD, HMD, GMD, and naive fusion, respectively, which shows that the uncertainty in AMD fusion is very high if the proximity between individual means is large. The global observability in this scenario is highly dependent on the quality of feedback, since the local nodes are static. Due to a highly uncertain feedback, the quality of the local estimate does not improve over time, and the fused estimate fails to converge. Amalgamation for unobservable sensors and maneuvering targets is hence avoided due to the over-conservativeness of the resulting density, and inflation of variance is observed.

A similar trend is shown in the velocity RMSE plot in Fig. 2.14b where the AMD takes around 75 time steps to settle down. Harmonic mean density again proves to be the best technique among the mentioned conservative fusion strategies, with the lowest RMSE in both position and velocity. This is due to a tight approximation of ‘*common information*’ which produces a conservative density with the least variance while still being consistent. When compared to the naive fusion strategy, the average improvement is roughly 2.5 times in position and around 1.5 times in velocity.

For consistency, the average NEES over 500 Monte-Carlo runs is plotted in Fig. 2.13b. Since the fused densities are Gaussian mixtures, the average NEES are calculated using the Gaussian approximation. One-sided 95% probability is taken as the threshold, which is shown by a thin blue line in the plot. It can be seen that apart



from one point in the initial transient region, there is no marker outside the threshold for 300 time steps which is quite acceptable. Also, the NEES for the proposed HMD implementation is closest to that of centralized filters. Note that the central filters are globally optimal (neglecting non-linear measurements), which proves the effectiveness of the proposed fusion method. The naive fusion results are totally inconsistent, as all points are outside the confidence region. Again, the arithmetic averaging proves to be overly conservative with the least average NEES.

The resulting tracks are shown in Fig. 2.15 where the AMD track has been omitted due to high track-loss. Since the naive density results in a lower non-consistent covariance, increasing the process noise improves track quality, since it has a similar effect as using a fudge factor [14]. The difference is evident in Fig. 2.15a and 2.15b. The GMD track (Fig. 2.15c) and the HMD track (Fig. 2.15d) faithfully track the target at a lower process noise.

Table 2.2: Parameters : Scenario 2

Parameter	Value
$\Delta T$	1 second
$R_1$	$(1.5^\circ)^2$
$R_2$	$(2^\circ)^2$
$\tilde{q}_{NCV}$	$10^{-2} \text{ m}^2/\text{sec}^3$
$\tilde{q}_{NCA}$	$10^{-3} \text{ m}^2/\text{sec}^5$
IMM transition matrix	$\begin{bmatrix} 0.8 & 0.2 \\ 0.8 & 0.2 \end{bmatrix}$

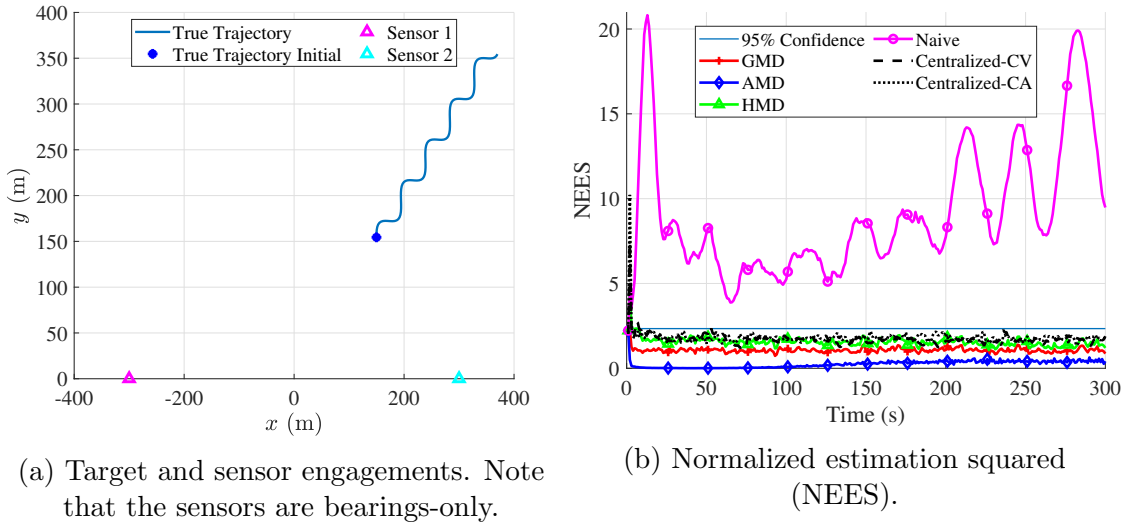


Figure 2.13: Scenario 2.

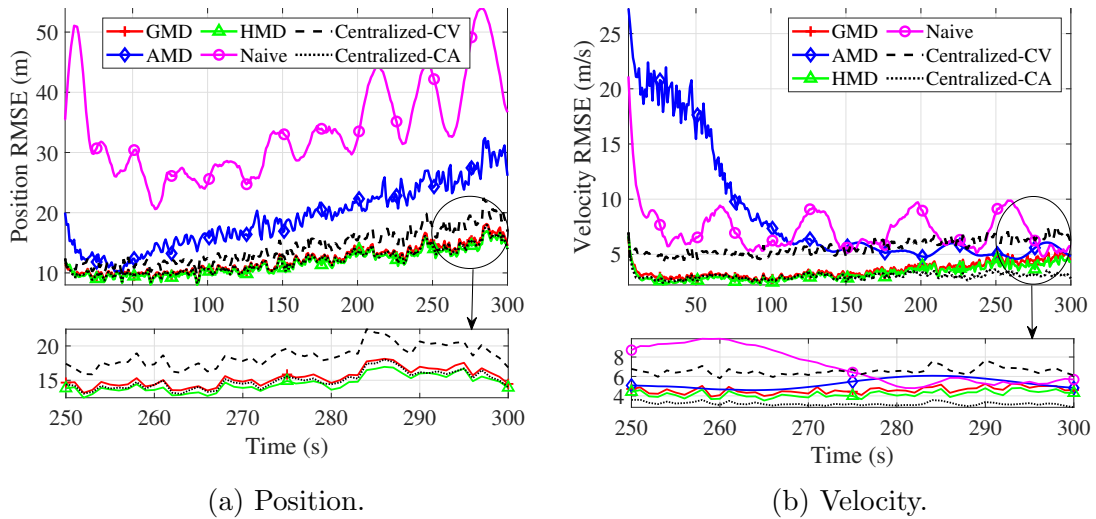


Figure 2.14: Root mean square error performance : Scenario 2

## 2.8 Conclusion

In this chapter, a new conservative strategy that could fuse both uni-modal and multi-modal local track densities without any additions to the framework has been presented. The proposed strategy has been proven to be effective in non-independent

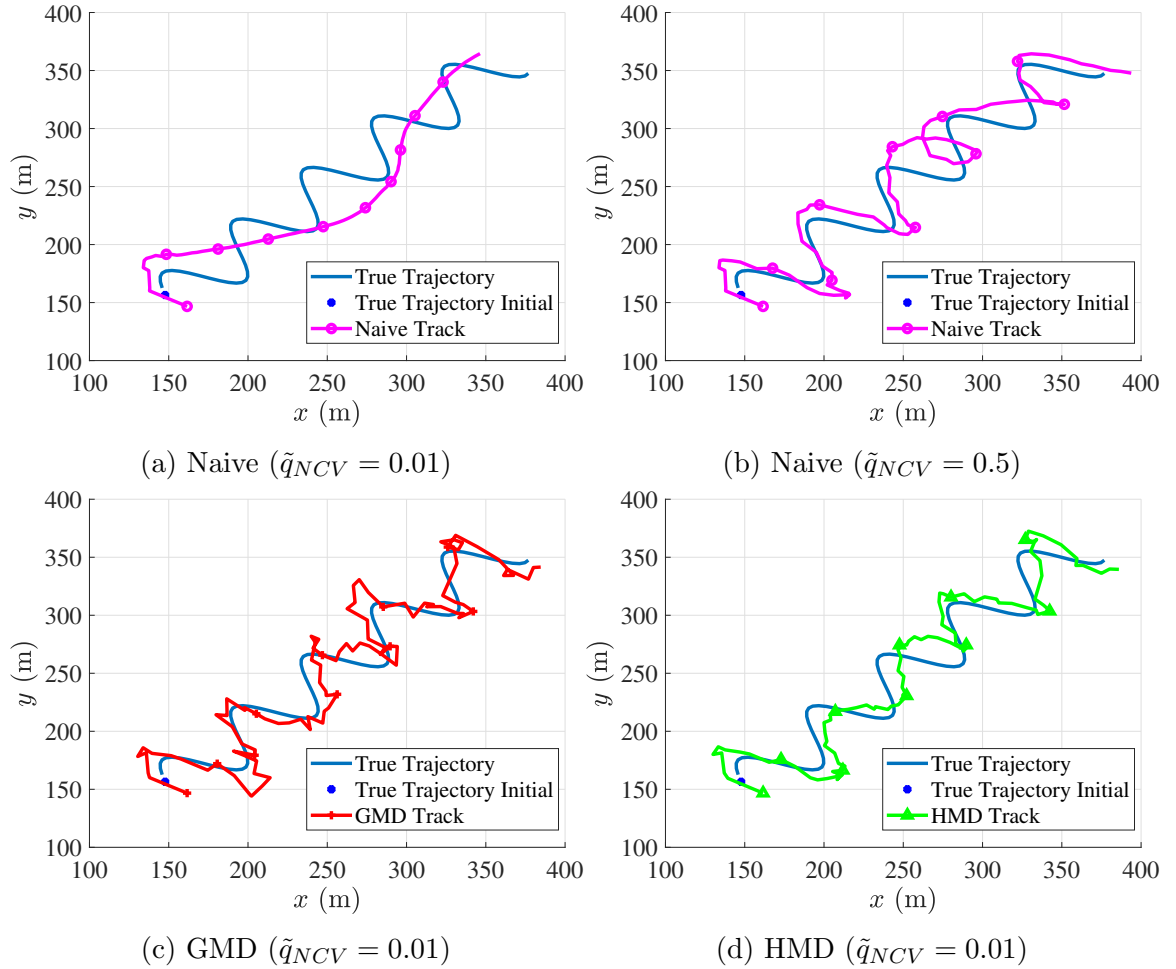


Figure 2.15: Fused tracks with various approaches for scenario 2. The arithmetic mean density has been excluded due to very high track-loss.

fusion architectures, both theoretically and in simulations. We investigated the mathematical properties and derived some useful equations required in the development of a viable fusion method. It was proven that the proposed harmonic mean density does not double count information and has a higher entropy than GMD, proving its conservativeness. It can be seen that the higher entropy is not entirely due to uncertainty but is due to the inclusion of the normalization constant in the formulation,

which is also proved to be smaller than that of GMD. Moreover, the approximate implementation of the proposed method is also presented. Analysis shows that HMD is as computationally effective as other conservative strategies existing in the literature. The efficacy of the harmonic mean density is presented in two real-life scenarios and shown to have the least RMSE among the discussed conservative fusion strategies. The NEES results show that the HMD is consistent and is not overly conservative like the arithmetic mean density. A few existing research areas discuss the evaluation of fusion weights and perhaps an accurate implementation without using Gaussian approximation for the denominator in equation (2.6.1). Nevertheless, the HMD is a promising candidate for the fusion of correlated estimates in distributed or decentralized network architectures.

# Chapter 3

## Harmonic Mean Density Fusion : Performance and Comparison

### 3.1 Abstract

A distributed sensor fusion architecture is preferred in a real target-tracking scenario as compared to a centralized scheme since it provides many practical advantages in terms of computation load, communication bandwidth, fault-tolerance, and scalability. In multi-sensor target-tracking literature, such systems are better known by the pseudonym – track fusion, since processed tracks are fused instead of raw measurements. A fundamental problem, however, in such systems is the presence of unknown correlations between the tracks which renders using a standard Kalman filter useless.

A widely-accepted solution is covariance intersection (CI) which provides near-optimal estimates but at the cost of a highly conservative covariance. Thus, the estimates are pessimistic which results in a delayed error convergence. Also, fusion of Gaussian mixture densities is an active area of research where standard methods of

track fusion cannot be used. In this chapter, harmonic mean density (HMD) based fusion is discussed which seems to handle both of these issues. We present insights on HMD fusion and prove that the method is a result of minimizing average Pearson divergence. This chapter also provides an alternative and easy implementation based on an importance-sampling like method without the requirement of a proposal density. Similarity of HMD with inverse covariance intersection is an interesting find, and have been discussed in detail.

Results based on a real-world multi-target multi-sensor scenario show that the proposed approach converges quickly than existing track fusion algorithms while also being consistent as evident from the normalized estimation-error squared (NEES) plots.

## 3.2 Introduction

Even though sensor technologies are reaching new heights in terms of accuracy, size, processing time, and portability, it is still not prudent to depend on a single sensor for large-scale systems. A multi-sensor system provides unmatched benefits, including accuracy, fault tolerance, scalability, and, in some cases, observability. In the widely researched domain of multiple target tracking, the employed sensors have a limited field of view (FOV) which constrains their applicability. Thus, to scale the area of observation, multiple sensors are deployed, which transmit their observations either in the form of raw measurements (centralized) or process the raw data before sending the resulting processed observations (tracks) to a fusion facility. Such networked architectures are highly employed in systems such as simultaneous localization and mapping (SLAM), autonomous vehicles, air traffic control, adversary interception,

underwater surveillance, and others.

The architecture wherein raw observations are processed locally using standard techniques like a Kalman filter and sent to the nearest fusion center is the focus of this chapter. Such an architecture is better suited to cater to multi-sensor target tracking problems since it offers numerous advantages in comparison to a centralized mechanism as discussed in the last chapter.

There are two major classes of solution preferred for the case of track-fusion – 1. Bayesian methods, 2. Pooling methods. The Bayesian methods refer to the solution where Bayes' theorem is employed in some sense, either after decorrelating the local tracks or by computing a globalized likelihood function using local node parameters. It is also possible to compute the exact value of cross-correlation between any two local tracks using the recursion provided in [9], but this would require knowledge of the local Kalman gain at each fusion instant and also memory for storing  ${}^n C_2$  number of cross-correlations at the fusion center for  $n$  nodes. Such methods include the optimal T2TF algorithm, information matrix fusion, and tracklet fusion. Using a standard Kalman filter by ignoring cross-correlation is usually termed naive fusion.

The pooling methods combine probability density functions directly using mixing weights such that some criterion is minimized. Under the target-tracking domain, such methods include the famous covariance intersection and its variants, as well as its generalized form known as the geometric mean density (GMD) fusion. These methods are preferred due to their relative ease of implementation in the Gaussian case. The ease of implementation stems from the fact that these methods do not inquire into the nature of cross-correlation. Thus, extra information from the local node, viz. Kalman gain, prior estimates from the last fusion, or sensor characteristics

are not required for fusion (at least for the ideal case). Such methods include the covariance intersection (CI) and its variants. A new class of pooling method known as harmonic mean density (HMD) based track fusion was introduced by the authors in [84], which is also the focus of this chapter.

### 3.2.1 Covariance Intersection and Variants

For a known cross-covariance, the resulting fused covariance ellipsoid can be shown to lie in the intersection between the local covariance ellipsoids [50, 27]. Thus, a convex combination of the covariance and mean of the estimates should yield a consistent fused estimate [50], which was utilized in the formulation of the original covariance intersection algorithm. In the presence of unknown cross-correlations, CI constitutes an upper bound on the cross-correlation, which makes it conservative. Later, many variants have been proposed to increase the tightness of this upper bound.

One such variant is the split-covariance intersection filter (SCIF), which assumes that a single local track is a package consisting of independent and dependent estimates separately, such that the local estimate is a sum of independent and dependent estimates. The algorithm then operates on these components in a way similar to the covariance intersection algorithm. The result is the fused estimates with their dependent and correlated components. The existence of a correlation such that independent and dependent components sum up to form a track is, however, vague. Also, the independent parts of a track are impossible to identify. If both components are available explicitly at the fusion center, then an optimal algorithm could be possible, according to the author's knowledge.

The ellipsoidal intersection (EI) in [85] aims to separate the mutual and exclusive



information contents from a local estimate and fuse them such that the mutual information is only counted once. The update equation resembles naive fusion before subtracting the mutual-information component. An interesting feature is that the algorithm computes the maximum possible cross-correlation ellipsoid, which results in a lower fused covariance relative to CI. However, it was shown in [73, 74] that EI can result in non-consistent estimates. The paper also proposed an additional CI variant, termed inverse-covariance intersection (ICI), which claims consistency of the fused estimate in addition to providing a less-conservative fused covariance relative to CI. However, both papers on ICI have not provided any effect on the root-mean-square error. Also, both CI and ICI need to compute a one-dimension minima search for the fusion parameter  $\omega$ . Though the cost function in both algorithms is convex, the one in ICI requires inverting 3 matrices compared to 2 in CI, thus being slightly slower. Also, no alternatives to ICI, EI, and SCIF for densities other than Gaussian exist in the current literature.

Harmonic mean density (HMD) based track fusion is a viable solution to both problems, including highly conservative fused covariance and the fusion of Gaussian mixtures. In this context, the contributions of this chapter are as follows :

- We prove that, like the GMD, which is the result of minimizing average Kullback-Leibler divergence (KLD), the HMD minimizes average Pearson  $\chi^2$  divergence and, thus, the reverse Neyman  $\chi^2$  divergence.
- The chapter presents important developments on the consistency and unbiasedness of fused estimates.
- Similarity of HMD implementation has been shown towards inverse covariance intersection (ICI). An extension of ICI towards Gaussian mixtures has been

presented.

- We provide an alternate implementation of the HMD based on importance sampling without the requirement of a proposal density.
- A simple optimization approach to calculating the fusion weights has also been provided. The approach requires one matrix inversion and, hence, performs faster.
- It has been shown that the existing strategy for calculation of fusion weights in CI and ICI fails in the case of scalar estimates. Through an example, the superiority of HMD in the fusion of such cases has been shown.

The rest of the chapter is organized as follows: In Section 3.3, the properties of HMD fusion are reviewed along with standard track fusion techniques. Here, the existing implementation for HMD, where the Gaussian mixture denominator is approximated by a Gaussian density, is presented. We present a consistency analysis of this implementation in Section 3.4. The relationship between ICI and HMD has also been provided in this section, along with some comments on ICI. Section 3.5 provides a new sample-based implementation of HMD and GMD using importance sampling-like techniques. The comparison and results on multiple track-fusion scenarios are presented in Section 3.6. Finally, the chapter is concluded in Section 3.7.

### 3.3 Harmonic Mean Density Fusion

Given any two probability density functions,  $p_1(\mathbf{x})$  and  $p_2(\mathbf{x})$ , an interpolation between them can be constructed using an  $\omega$ -weighted generalized mean,  $\mathcal{M}_\omega(p_1(\mathbf{x}), p_2(\mathbf{x}))$ ,

also known as the Fréchet mean. Such generalized means can also be constructed as a result of optimizing a unique  $f$ -divergence measure [54],

$$\mathcal{M}_\omega(p_1(\mathbf{x}), p_2(\mathbf{x})) = \arg \min_{m(\mathbf{x})} \sum_{i=1}^2 \omega_i D_f [m(\mathbf{x}) || p_i(\mathbf{x})] \quad (3.3.1)$$

where  $D_f(\cdot)$  is a  $f$ -divergence [106], defined by,

$$D_f[p(\mathbf{x}) || q(\mathbf{x})] = \int q(\mathbf{x}) f\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right) dx \quad (3.3.2)$$

with probability densities  $p(\mathbf{x})$  and  $q(\mathbf{x})$ .  $f(t)$  is any convex function that is defined for  $t > 0$ , with  $f(1) = 0$  [33]. The resulting Fréchet mean is in general a mixture probability distribution, which when normalized results in a valid probability density function known as the  $\omega$ -weighted M-mixture,

$$\mathbf{M}_\omega(p_1(\mathbf{x}_k), p_2(\mathbf{x}_k)) = \frac{\mathcal{M}_\omega(p_1(\mathbf{x}_k), p_2(\mathbf{x}_k))}{\zeta_{\mathcal{M}_\omega}(p_1(\mathbf{x}_k), p_2(\mathbf{x}_k))}, \quad (3.3.3)$$

where  $\zeta_{\mathcal{M}_\omega}(p_1(\mathbf{x}_k), p_2(\mathbf{x}_k))$  is the normalization constant resulting from interpolation of  $p_1(x)$  and  $p_2(x)$  to form an abstract mean  $\mathcal{M}_\omega(\cdot)$ ,

$$\zeta_{\mathcal{M}_\omega}(p_1(\mathbf{x}_k), p_2(\mathbf{x}_k)) = \int_{\mathbb{R}^n} \mathcal{M}_\omega(p_1(\mathbf{x}_k), p_2(\mathbf{x}_k)) d\mathbf{x} \quad (3.3.4)$$

In the case of HMD, we take the abstract mean as the harmonic mean of participating local track densities. The resulting fused M-mixture  $p_f(\mathbf{x})$  is,

$$\frac{1}{p_f(\mathbf{x})} \propto \mathcal{M}_\omega^h(p_1(\mathbf{x}), p_2(\mathbf{x})) = \frac{\omega}{p_1(\mathbf{x})} + \frac{1-\omega}{p_2(\mathbf{x})} \quad (3.3.5)$$

$$\implies p_f(\mathbf{x}) \propto \frac{p_1(\mathbf{x})p_2(\mathbf{x})}{(1-\omega)p_1(\mathbf{x}) + \omega p_2(\mathbf{x})} \quad (3.3.6)$$

Where the proportionality constant is the normalization factor  $1/\zeta_{\mathcal{M}_\omega^h}(\cdot)$ . Note that the mixture density in the denominator prohibits closed-form solutions, even for the Gaussian case. Even with simple approximations, HMD-based track fusion was found to perform exceptionally well compared to other track fusion strategies. Many important properties of HMD akin to track fusion have been proven by the authors in [84]. Some of these are,

- (i) HMD avoids double counting of information, unlike naive fusion in (2.3.2).

Assuming conditional dependence, such that

$$p(\mathbf{z}^1|\mathbf{x}) = p(\mathbf{z}^{1/2}|\mathbf{x})p(\mathbf{z}^1 \cap \mathbf{z}^2|\mathbf{x}), \quad (3.3.7)$$

The notation  $p(\mathbf{z}^{1/2}|\mathbf{x})$  denotes the exclusive information present with sensor track 1 relative to 2 such that  $p(\mathbf{z}^{1/2}|\mathbf{x}) \cap p(\mathbf{z}^{2/1}|\mathbf{x}) = \emptyset$ . It is easy to observe that using HMD,

$$\begin{aligned} p(\mathbf{x}|\mathbf{z}^1 \cup \mathbf{z}^2) &\propto \frac{p(\mathbf{z}^{1/2}|\mathbf{x})p(\mathbf{z}^{2/1}|\mathbf{x})}{(1-\omega)p(\mathbf{z}^{1/2}|\mathbf{x}) + \omega p(\mathbf{z}^{2/1}|\mathbf{x})} \\ &\times p(\mathbf{z}^1 \cap \mathbf{z}^2|\mathbf{x})p(\mathbf{x}). \end{aligned} \quad (3.3.8)$$

Thus, the common information  $p(\mathbf{z}^1 \cap \mathbf{z}^2 | \mathbf{x})$  is only accounted once.

- (ii) HMD has the potential to be a recursive fusion strategy since it is an abstract mean. To illustrate, let  $p_1$ ,  $p_2$  and  $p_3$  be three local densities. Then, it can be proved that

$$\mathcal{M}_\omega^h(p_1, p_2, p_3) = \mathcal{M}_\omega^h\left(\mathcal{M}_\omega^h(p_1, p_2), p_3\right), \quad (3.3.9)$$

which makes it easier to fuse local tracks pairwise.

- (iii) The second derivative of the normalization constant with respect to  $\omega$  in case of HMD can be proved to be always positive for valid local densities  $p_1(\mathbf{x})$  and  $p_2(\mathbf{x})$ . This suggests a convex nature of the normalization constant, like in the case of GMD. It was also found that the normalization constant in the case of HMD is less than that of GMD.

$$\int_{\mathbb{R}^n} \frac{p_1(\mathbf{x})p_2(\mathbf{x})}{\omega_2 p_1(\mathbf{x}) + \omega_1 p_2(\mathbf{x})} \leq \int_{\mathbb{R}^n} \frac{1}{p_1^{\omega_1}(\mathbf{x})p_2^{\omega_2}(\mathbf{x})} \quad (3.3.10)$$

Due to this property, the entropy of HMD can be proved to be less than that of GMD when the densities are properly normalized. Thus, HMD produces relatively tight estimates.

- (iv) Since HMD is an abstract mean, it is bounded from above and below by participating densities. It ensures that the fusion can be performed unlimited number of times, and the worse result we can get is the infimum of the set of participating densities.

A lot of work on conservative fusion in contemporary literature is focused on how

to reduce the over-conservativeness of covariance intersection and its generalization. It was shown in simulations of [84] that the HMD does not suffer from over-inflated fused covariance, resulting in a superior accuracy. Also, in the simulations, the normalized estimation error squared plots proved the consistency of HMD in such scenarios.

### 3.3.1 Average Divergence Minimization

Conservative fusion techniques like arithmetic and geometric averaging construct a fused density by creating a mutual agreement between the participating local densities. This mutual agreement is based on minimizing a metric between the output density and each of the local density functions. Therefore, the techniques existing in the literature are found to be a result of minimizing some average divergence measure.

- The covariance intersection and its generalization are a result of minimizing  $\omega$ -weighted average KL divergence.
- The arithmetic average density is a result of minimizing  $\omega$ -weighted reverse KL divergence.

While in [84], it was proved that the HMD has the potential to become a Kullback-Leibler average density. Here, we prove that it actually results in a minimization of the average Pearson  $\chi^2$  divergence. Though, the divergence minimization in any of the pooling techniques does not affect the fusion accuracy in any sense, it provides an intuition about the geometry of the information process that the local density undergoes during fusion.

**Theorem 6.** *The harmonic mean density between  $p_1(\mathbf{x})$  and  $p_2(\mathbf{x})$  minimizes the  $\omega$ -weighted average Pearson  $\chi^2$  divergence, and thus, the reverse Neyman  $\chi^2$  divergence.*

*Proof.* Denote the  $\omega$ -weighted average Pearson  $\chi^2$  divergence by  $J$ , and the HMD by  $p_h(\mathbf{x})$ , then,

$$J(p_h(\mathbf{x})) = \sum_{i=1}^2 \omega_i D_{\chi_P^2}(p_h(\mathbf{x}) || p_i(\mathbf{x})) = \sum_{i=1}^2 \omega_i \frac{1}{2} \int_{\infty}^{\infty} \frac{(p_h(\mathbf{x}) - p_i(\mathbf{x}))^2}{p_i(\mathbf{x})} d\mathbf{x} \quad (3.3.11)$$

where  $D_{\chi_P^2}$  stands for Pearson  $\chi^2$  divergence. Note that there are two constraints on  $p_h(\mathbf{x})$ , which are — 1.  $p_h(\mathbf{x}) \geq 0$ , and 2.  $\int_{\mathbb{R}^n} p_h(\mathbf{x}) = 1$ . In order to proceed with minimization using calculus of variations, we ignore the constraints and simply minimize the functional. The minimized function will then be checked for validity of a proper probability density function.

The condition of minimization is upon the first variation as,

$$\delta J(p_h(\mathbf{x})) = \frac{d}{dt} \left( J(p_h(\mathbf{x}) + t g_h(\mathbf{x})) \right)_{t=0} = 0 \quad (3.3.12)$$

Reversing the summation and integral in eqn. (3.3.11), and substituting in eqn. (3.3.12),

$$\begin{aligned} \delta J(p_h(\mathbf{x})) &= \int_{\infty}^{\infty} \frac{1}{2} \sum_{i=1}^2 \omega_i \frac{d}{dt} \frac{(p_h(\mathbf{x}) + t g_h(\mathbf{x}) - p_i(\mathbf{x}))^2}{p_i(\mathbf{x})} d\mathbf{x} \\ &= \int_{\infty}^{\infty} \frac{1}{2} \sum_{i=1}^2 \omega_i \frac{2(p_h(\mathbf{x}) + t g_h(\mathbf{x})) g_h(\mathbf{x}) - 2 p_i(\mathbf{x}) g_h(\mathbf{x})}{p_i(\mathbf{x})} \\ &= \int_{\infty}^{\infty} \sum_{i=1}^2 \omega_i \left[ \frac{(p_h(\mathbf{x}) + t g_h(\mathbf{x}))}{p_i(\mathbf{x})} - 1 \right] g_h(\mathbf{x}) = 0 \end{aligned} \quad (3.3.13)$$

Substituting  $t = 0$  for the first variation, we get the following equation for  $p_h(\mathbf{x})$

$$p_h(\mathbf{x}) \sum_{i=1}^2 \frac{\omega_i}{p_i(\mathbf{x})} = \sum_{i=1}^2 \omega_i = 1 \quad (3.3.14)$$

which implies,

$$p_h(\mathbf{x}) = \frac{1}{\sum_{i=1}^2 \frac{\omega_i}{p_i(\mathbf{x})}} = \frac{p_1(\mathbf{x})p_2(\mathbf{x})}{\omega_2 p_1(\mathbf{x}) + \omega_1 p_2(\mathbf{x})} \quad (3.3.15)$$

In general, it can be proved that,

$$p_h(\mathbf{x}) = \frac{1}{\sum_{i=1}^N \frac{\omega_i}{p_i(\mathbf{x})}} \quad (3.3.16)$$

where  $N$  is the number of local densities.

Since the participating local densities are valid pdfs, the first constraint for positivity is satisfied. The second constraint can be forced by normalizing the resulting density with  $\zeta^h = \int p_h(\mathbf{x})d(\mathbf{x})$ . The second variation can be shown to be positive, which proves that the result is actually a minima.

$$\begin{aligned} \delta^2 J(p_h(\mathbf{x})) &= \frac{d^2}{dt^2} \left( J(p_h(\mathbf{x}) + tg_h(\mathbf{x})) \right)_{t=0} \\ &= \int_{-\infty}^{\infty} \sum_{i=1}^2 \omega_i \frac{g_h(\mathbf{x})^2}{p_i(\mathbf{x})} d\mathbf{x} \geq 0. \end{aligned} \quad (3.3.17)$$

For proving the case for  $D_{\chi_N^2}(\cdot)$ , the reverse Neyman divergence, it can be noted that both  $D_{\chi_N^2}(\cdot)$  and  $D_{\chi_P^2}(\cdot)$  are special cases of  $\alpha$ -divergence  $D_\alpha(\cdot)$  which satisfy the



property,

$$D_{\chi_P^2}(\cdot) = D_{(\alpha=2)}(p_i(\mathbf{x})||p_j(\mathbf{x})) = D_{(\alpha=-1)}(p_j(\mathbf{x})||p_i(\mathbf{x})) = D'_{\chi_N^2}(\cdot) \quad (3.3.18)$$

where  $D'(\cdot)$  stands for divergence obtained by reversing the arguments.

Note that the  $\chi^2$  divergence between Gaussian densities only exists upon a certain condition among the covariances. It can be easily proved that this condition is satisfied for the fused density and participating local density. Hence the average divergence in eqn. (3.3.11) will always exist. The proof is left for the readers.  $\square$

### 3.3.2 Implementation using Gaussian Approximation

The simplest implementation of HMD takes the following form, when the denominator in a Gaussian mixture is replaced by a Gaussian equivalent [84]. Let's call this implementation HMD-GA, which is given by (dropping time-step  $k$  for brevity),

$$\frac{1}{\zeta_{\mathcal{M}_w^h} \omega_2 p_1(\mathbf{x}) + \omega_1 p_2(\mathbf{x})} \propto \frac{p_1(\mathbf{x})p_2(\mathbf{x})}{\mathcal{N}(\mathbf{x}; \boldsymbol{\gamma}_m^h, \boldsymbol{\Gamma}_m^h)} \quad (3.3.19)$$

which gives the fused mean and covariance as,

$$\boldsymbol{\Gamma}^f = \left( \boldsymbol{\Gamma}_1^{-1} + \boldsymbol{\Gamma}_2^{-1} - \boldsymbol{\Gamma}_m^{h-1} \right)^{-1}, \quad (3.3.20a)$$

$$\hat{\mathbf{x}}^f = \boldsymbol{\Gamma}^f \left[ \boldsymbol{\Gamma}_1^{-1} \hat{\mathbf{x}}_1 + \boldsymbol{\Gamma}_2^{-1} \hat{\mathbf{x}}_2 - \boldsymbol{\Gamma}_m^{h-1} \boldsymbol{\gamma}_m^h \right], \quad (3.3.20b)$$

where  $\boldsymbol{\gamma}_m^h$  and  $\boldsymbol{\Gamma}_k^h$  are obtained using the Gaussian approximation of the mixture. These quantities represent the mutual information due to cross-correlations, which is subtracted once in (3.3.20). The implementation is simple, intuitive, and requires

three matrix inversions, making it of the order  $\mathcal{O}(n^3)$ , where  $n$  is the dimension of state estimates. This is equivalent to all other fusion algorithms.

### 3.4 Consistency Analysis of HMD-GA

Using a Gaussian approximation for the mixture distribution in the denominator of eqn. (3.3.6), we need to evaluate the consistency of our estimate. Even though the simulations suggest HMD as a consistent fusion method, the spread-of-means term in Gaussian approximation might pose a problem in convergence. In this section, we present the conditions of consistency of HMD-GA in a general case.

Comparing the construction of HMD-GA in eqn. (3.3.20) with that of the optimal fusion equation in eqn. (4.3.3), it is easy to see that the mutual information component in HMD ( $\boldsymbol{\gamma}_m^h, \boldsymbol{\Gamma}_m^h$ ) is the Gaussian equivalent of the mixture,

$$\boldsymbol{\gamma}_m^h = \omega_2 \hat{\mathbf{x}}_1 + \omega_1 \hat{\mathbf{x}}_2 \quad (3.4.1)$$

$$\boldsymbol{\Gamma}_m^h = \omega_2 \boldsymbol{\Gamma}_1 + \omega_1 \boldsymbol{\Gamma}_2 + \tilde{\boldsymbol{\Gamma}} \quad (3.4.2)$$

where  $\tilde{\boldsymbol{\Gamma}}$  is the spread-of-means term, given by,

$$\tilde{\boldsymbol{\Gamma}} = \sum_{i=1}^2 (1 - \omega_i) [\hat{\mathbf{x}}_i - \boldsymbol{\gamma}_m^h] [\hat{\mathbf{x}}_i - \boldsymbol{\gamma}_m^h]^T. \quad (3.4.3)$$

Which can be manipulated and alternatively written as,

$$\tilde{\boldsymbol{\Gamma}} = \omega_1 \omega_2 [\hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_2] [\hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_2]^T \quad (3.4.4)$$

Note that  $\tilde{\boldsymbol{\Gamma}}$  is a rank-one matrix, with an eigen-value  $\omega_1 \omega_2 [\hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_2]^T [\hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_2]$  with

multiplicity 1, and the rest of the eigen-values zero with multiplicity  $n - 1$ ;  $n$  being the dimension of the state.

As per the definition of consistency, the estimate error with respect to the true state of the target should match the covariance presented by the estimator. Mathematically, the expected error should converge to its covariance matrix. If  $(\hat{\mathbf{x}}, \mathbf{\Gamma})$  is an unbiased estimate pair, then it is needed that,

$$\mathbb{E} [\mathbf{x} - \hat{\mathbf{x}}] [\mathbf{x} - \hat{\mathbf{x}}]^T = \mathbf{\Gamma}, \quad (3.4.5)$$

at least asymptotically. Since in our problem the correlations between estimates are missing, it is impossible to match eqn. (3.4.5), thus, we resort to the definition of consistency as conservativeness in eqn. (2.3.6). Therefore, we are looking for a pessimistic estimate of error covariance. As the current literature does, we will use the words conservativeness and consistency interchangeably, even though this is not true.

To analyze consistency, we first prove that HMD-GA provides an unbiased estimate. Taking expectation on both sides of eqn. (3.3.20b), we get,

$$\mathbb{E} [\hat{\mathbf{x}}^f] = \mathbf{\Gamma}^f \left[ \mathbf{\Gamma}_1^{-1} \mathbb{E} [\hat{\mathbf{x}}_1] + \mathbf{\Gamma}_2^{-1} \mathbb{E} [\hat{\mathbf{x}}_2] - \mathbf{\Gamma}_m^{h-1} \mathbb{E} [\boldsymbol{\gamma}_m^h] \right], \quad (3.4.6)$$

The local estimates are assumed unbiased which means  $\mathbb{E}[\hat{\mathbf{x}}^1] = \mathbb{E}[\hat{\mathbf{x}}^2] = \mathbb{E}[\boldsymbol{\gamma}_m^h] = \mathbf{x}$ . Thus,

$$\mathbb{E} [\hat{\mathbf{x}}^f] = \mathbf{\Gamma}^f (\mathbf{\Gamma}^f)^{-1} \mathbf{x} = \mathbf{x} \quad (3.4.7)$$

wherein eqn. (3.3.20a) has been employed. Next, we calculate the estimation error by subtracting eqn. (3.3.20b) from eqn. (3.4.6) (using the notation  $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$ ),

$$\tilde{\mathbf{x}}^f = \mathbf{\Gamma}^f \left[ \mathbf{\Gamma}_1^{-1} \tilde{\mathbf{x}}_1 + \mathbf{\Gamma}_2^{-1} \tilde{\mathbf{x}}_2 - \mathbf{\Gamma}_m^{h^{-1}} \tilde{\boldsymbol{\gamma}}_m^h \right] \quad (3.4.8)$$

Our requirement becomes,

$$\mathbf{\Gamma}^f - \mathbb{E} \left[ \tilde{\mathbf{x}}^f (\tilde{\mathbf{x}}^f)^T \right] \succeq \mathbf{0} \quad (3.4.9)$$

which leads us to,

$$\begin{aligned} (\mathbf{\Gamma}^f) - \mathbf{\Gamma}^f & \left[ \mathbf{\Gamma}_1^{-1} + \mathbf{\Gamma}_2^{-1} + \mathbf{\Gamma}_1^{-1} \mathbf{\Gamma}_{12} \mathbf{\Gamma}_2^{-1} + \mathbf{\Gamma}_2^{-1} \mathbf{\Gamma}_{21} \mathbf{\Gamma}_1^{-1} \right. \\ & - (\omega_2 \mathbf{\Gamma}_1^{-1} \mathbf{\Gamma}_{12} + \omega_1 \mathbf{\Gamma}_2^{-1} \mathbf{\Gamma}_{21}) \mathbf{\Gamma}_m^{h^{-1}} + \mathbf{\Gamma}_m^{h^{-1}} \\ & \left. + \mathbf{\Gamma}_m^{h^{-1}} (\omega_2 \mathbf{\Gamma}_{21} \mathbf{\Gamma}_1^{-1} + \omega_1 \mathbf{\Gamma}_{12} \mathbf{\Gamma}_2^{-1}) \right] \mathbf{\Gamma}^f \succeq \mathbf{0} \end{aligned} \quad (3.4.10)$$

where  $\mathbf{\Gamma}_{12}$  is the cross-correlation matrix between the estimation error of  $\hat{\mathbf{x}}_1$  and  $\hat{\mathbf{x}}_2$ . It has also been assumed that  $\boldsymbol{\gamma}_m^h, \mathbf{\Gamma}_m^h$  form a consistent pair, such that  $\mathbb{E} \left[ \tilde{\boldsymbol{\gamma}}_m^h (\tilde{\boldsymbol{\gamma}}_m^h)^T \right] = \mathbf{\Gamma}_m^h$ . Multiplying both sides by  $\mathbf{\Gamma}^{f^{-1}}$  and using eqn. (3.3.20a), we get,

$$\begin{aligned} & \left( \omega_2 \mathbf{\Gamma}_1^{-1} \mathbf{\Gamma}_{12} \mathbf{\Gamma}_m^{h^{-1}} + \omega_1 \mathbf{\Gamma}_m^{h^{-1}} \mathbf{\Gamma}_{12} \mathbf{\Gamma}_2^{-1} - \mathbf{\Gamma}_1^{-1} \mathbf{\Gamma}_{12} \mathbf{\Gamma}_2^{-1} \right) + \\ & \left( \omega_1 \mathbf{\Gamma}_2^{-1} \mathbf{\Gamma}_{21} \mathbf{\Gamma}_m^{h^{-1}} + \omega_2 \mathbf{\Gamma}_m^{h^{-1}} \mathbf{\Gamma}_{21} \mathbf{\Gamma}_1^{-1} - \mathbf{\Gamma}_2^{-1} \mathbf{\Gamma}_{21} \mathbf{\Gamma}_1^{-1} \right) \succeq \mathbf{0} \end{aligned} \quad (3.4.11)$$

For positive semi-definiteness (PSD) of the left hand side, each of the term inside parenthesis should be PSD. Evaluating each term, we can find sufficient condition for

consistency. The first term on LHS states,

$$\omega_2 \mathbf{\Gamma}_1^{-1} \mathbf{\Gamma}_{12} \mathbf{\Gamma}_m^{h-1} + \omega_1 \mathbf{\Gamma}_m^{h-1} \mathbf{\Gamma}_{12} \mathbf{\Gamma}_2^{-1} - \alpha \mathbf{\Gamma}_1^{-1} \mathbf{\Gamma}_{12} \mathbf{\Gamma}_2^{-1} - (1 - \alpha) \mathbf{\Gamma}_1^{-1} \mathbf{\Gamma}_{12} \mathbf{\Gamma}_2^{-1} \succeq \mathbf{0} \quad (3.4.12)$$

for any  $\alpha \in [0, 1]$ . The final step is to aggregate similar terms and evaluate for positive-definiteness, which gives,

$$\omega_2 \mathbf{\Gamma}_m^{h-1} \succeq \alpha \mathbf{\Gamma}_1^{-1} \quad \text{and,} \quad \omega_2 \mathbf{\Gamma}_m^{h-1} \succeq (1 - \alpha) \mathbf{\Gamma}_2^{-1} \quad (3.4.13)$$

The resulting condition is,

$$\mathbf{\Gamma}_m^h \preceq \frac{\omega}{\alpha} \mathbf{\Gamma}_1 + \frac{1 - \omega}{1 - \alpha} \mathbf{\Gamma}_2 \quad (3.4.14)$$

where  $\alpha$  can be conveniently chosen as equal to  $\omega$  which produces the condition,

$$\mathbf{\Gamma}_m^h \preceq \mathbf{\Gamma}_1 + \mathbf{\Gamma}_2 \quad (3.4.15)$$

Same result can be produced from second block in eqn. (3.4.11). This upper limit follows for all fusion methods following the structure mentioned in eqn. (3.4.32) and using  $\gamma_m^h$ . It is easy to see that in naive fusion,  $\mathbf{\Gamma}_m^h \rightarrow \infty$ , producing an inconsistent estimate.

We can interpret the relation using the linear correlation coefficient  $\rho$ , which takes positive values in the case of homogeneous track fusion. When  $\rho = 0$ , it is obvious that the estimates are independent and HMD-GA is therefore consistent. The case of perfect correlation,  $\rho = 1$  can be understood using the following assumption for the

local track estimate,

$$\hat{\mathbf{x}}_i = \mathbf{x} + \mathbf{v}_i \quad (3.4.16)$$

where  $\mathbf{x}$  is the true value and the corresponding noise term,  $\mathbf{v}_i$  is sampled from the distribution  $\mathcal{N}(\mathbf{0}, \mathbf{\Gamma}_i)$ . This simply follows from the unbiased-ness and consistency of the local tracker. In case of perfect correlation, the local estimates  $\hat{\mathbf{x}}_i$  and  $\hat{\mathbf{x}}_j$ , both following the relation in eqn. (3.4.16), should exhibit a linear relationship,

$$\hat{\mathbf{x}}_1 = \mathbf{A}\hat{\mathbf{x}}_2 + \mathbf{B} \quad (3.4.17)$$

It follows from unbiasedness that  $\mathbf{A} = \mathbf{I}$  and  $\mathbf{B} = \mathbf{0}$ , where  $\mathbf{I}$  is the identity matrix of appropriate order. Thus,  $\hat{\mathbf{x}}_1 = \hat{\mathbf{x}}_2$  should follow and HMD-GA effectively reduces to inverse covariance intersection which is proven as a consistent method in [74].

Replacing the spread-of-means term in eqn. (3.4.4) by its expected value,

$$\mathbb{E} [\tilde{\mathbf{\Gamma}}] = \omega_1\omega_2 [\mathbf{\Gamma}_1 + \mathbf{\Gamma}_2 - \mathbf{\Gamma}_{12} - \mathbf{\Gamma}_{21}] \quad (3.4.18)$$

then, the mutual information component subtracted by the bound given in eqn. (3.4.15) is given by,

$$\mathbb{E} [\mathbf{\Gamma}_m^h] - [\mathbf{\Gamma}_1 + \mathbf{\Gamma}_2] = \mathbf{\Gamma}_1 [\omega_2 + \omega_1\omega_2 - 1] + \mathbf{\Gamma}_2 [\omega_1 + \omega_1\omega_2 - 1] - \omega_1\omega_2 [\mathbf{\Gamma}_{12} + \mathbf{\Gamma}_{21}] \quad (3.4.19)$$

has to be negative-definite for consistency. This requires the cross-covariance matrices  $\mathbf{\Gamma}_{12}$  and  $\mathbf{\Gamma}_{21}$  to be positive definite. From the Bar-Shalom-Campo formula [9, 10], we

know that (at sample-time  $k$ ),

$$\mathbf{\Gamma}_k^{12} = (\mathbf{I} - \mathbf{K}_k^1 \mathbf{H}_k^1) \mathbf{F}_{k-1} \mathbf{\Gamma}_{k-1}^{12} \mathbf{F}_{k-1}^T (\mathbf{I} - \mathbf{K}_k^2 \mathbf{H}_k^2)^T + (\mathbf{I} - \mathbf{K}_k^1 \mathbf{H}_k^1) \mathbf{Q}_{k-1} (\mathbf{I} - \mathbf{K}_k^2 \mathbf{H}_k^2)^T \quad (3.4.20)$$

which proves that the cross-covariance matrices are indeed positive-definite (in the case of homogeneous case only). Here, the superscript denotes node,  $\mathbf{K}$  is the Kalman gain,  $\mathbf{H}$  is the measurement mapping matrix,  $\mathbf{F}$  is the state-transition matrix, and  $\mathbf{Q}$  is the process-noise covariance matrix. The subscript denotes time, and the superscript denotes node.

Therefore, as long as the spread-of-means term is around its expected value, HMD-GA provides consistent results. This strongly requires consistency of individual estimate.

Intuitively, for scalar estimates  $(\hat{x}_1, \sigma_1^2)$  and  $(\hat{x}_2, \sigma_2^2)$ , it follows that,

$$\tilde{x}_1 = \hat{x}_1 - x \leq \pm 1.96\sigma_1, \quad \text{with 95\% confidence} \quad (3.4.21)$$

and same for  $\hat{x}_2$ . Thus, the spread-of-means term in this case satisfies the following inequality (using the same confidence level),

$$[\hat{x}_1 - \hat{x}_2]^2 = [\tilde{x}_1 - \tilde{x}_2]^2 \leq 3.84 \max(\sigma_1^2, \sigma_2^2) \quad (3.4.22)$$

since due to positivity of  $\mathbb{E}[\tilde{x}_1 \tilde{x}_2^T]$ , both  $\hat{x}_1$  and  $\hat{x}_2$  need to be on the same side of their respective means. The conditions of consistency in this case, evaluated using

eqn. (3.4.15) is,

$$\max(\sigma_1^2, \sigma_2^2) \leq \frac{1}{3.84} \left[ \frac{\sigma_1^2}{\omega_1} + \frac{\sigma_2^2}{\omega_2} \right] \quad (3.4.23)$$

which asserts the consistency of HMD-GA for all practical values of  $\omega_1$ . The condition can be further relaxed by using a lower confidence level or a tighter bound in eqn. (3.4.22).

It can also be proved that without  $\tilde{\Gamma}$ ,  $\Gamma_m^h$  is consistent and exactly becomes inverse covariance intersection, which is discussed next.

### 3.4.1 Relation with ICI

Observing the implementation of HMD-GA in eqn. (3.3.20), its similarity with inverse covariance intersection (ICI) is obvious. ICI can be considered as the forced-consistent implementation of HMD-GA by removing the spread-of-means term in Gaussian approximation of the mixture. The main motivation behind using ICI is its consistency and a non-inflated covariance compared to covariance intersection. However, it doesn't enjoy the fruits of being a natural and generalized fusion algorithm like HMD and GMD. The fusion equation for ICI are,

$$\begin{aligned} \mathbf{\Gamma}^{\text{ICI}} &= [(\mathbf{\Gamma}_1)^{-1} + (\mathbf{\Gamma}_2)^{-1} - (\omega_1 \mathbf{\Gamma}_1 + \omega_2 \mathbf{\Gamma}_2)^{-1}]^{-1} \\ \hat{\mathbf{x}}^{\text{ICI}} &= \left[ (\mathbf{\Gamma}_1)^{-1} \hat{\mathbf{x}}_1 + (\mathbf{\Gamma}_2)^{-1} \hat{\mathbf{x}}_2 - (\omega_1 \mathbf{\Gamma}_1 + \omega_2 \mathbf{\Gamma}_2)^{-1} (\omega_1 \hat{\mathbf{x}}_1 + \omega_2 \hat{\mathbf{x}}_2) \right] \end{aligned} \quad (3.4.24)$$

with  $\omega_1 = (1 - \omega_2)$  as before. Note that the fusion weights have been interchanged in comparison to HMD-GA. This doesn't result in any loss of generalization.



### Correlation Structure Employed for ICI

In [74], a specific correlation structure was considered for the analysis and derivation of ICI. A general form of this structure including independent and non-independent data-sets are shown in Fig. 3.1 and Fig. 3.2

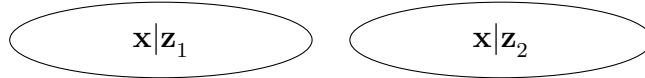


Figure 3.1: Independent Dataset

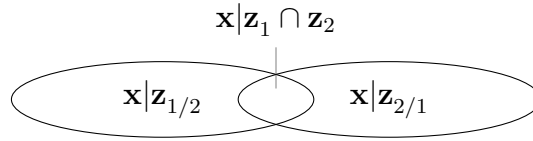


Figure 3.2: Non-independent Dataset [48]

Where the notation  $\mathbf{z}_{1/2}$  denotes measurement data contained in node 1 which is exclusive to node 2. Using the concept of conditional independence depicted in Fig. 3.2 and the Bayes' theorem, the conditional density based on common information can be written as,

$$\begin{aligned} P(\mathbf{x}|\mathbf{z}_1) &= P(\mathbf{x}|\mathbf{z}_{1/2})P(\mathbf{x}|\mathbf{z}_1 \cap \mathbf{z}_2) \\ \Rightarrow P(\mathbf{x}|\mathbf{z}_1 \cap \mathbf{z}_2) &= \frac{P(\mathbf{x}|\mathbf{z}_1)}{P(\mathbf{x}|\mathbf{z}_{1/2})} \end{aligned} \quad (3.4.25)$$

Incorporating the information from another sensor and using the fact that  $P(\mathbf{x}|\mathbf{z}_2 \cap \mathbf{z}_1) = P(\mathbf{x}|\mathbf{z}_1 \cap \mathbf{z}_2)$ , it can be concluded that,

$$P(\mathbf{x}|\mathbf{z}_1 \cap \mathbf{z}_2) = \frac{1}{2} \frac{P(\mathbf{x}|\mathbf{z}_1)}{P(\mathbf{x}|\mathbf{z}_{1/2})} + \frac{1}{2} \frac{P(\mathbf{x}|\mathbf{z}_2)}{P(\mathbf{x}|\mathbf{z}_{2/1})} \quad (3.4.26)$$

which is clearly a mixture density, similar to the mutual information component in the denominator of generalized HMD in eqn. (3.3.6).

The correlation structure mentioned here is directly in-line with that presented in [73, 74, 85] with the assumption of Gaussianity of all components. If  $P(\mathbf{x}|\mathbf{z}_1 \cap \mathbf{z}_2) = P(\mathbf{x}|\mathbf{z}_2 \cap \mathbf{z}_1)$  are assumed to be Gaussian distributed as  $\mathcal{N}(\mathbf{x}; \boldsymbol{\gamma}_m, \boldsymbol{\Gamma}_m)$ , and the independent component in a track  $P(\mathbf{x}|\mathbf{z}_{1/2})$  is Gaussian as  $\mathcal{N}(\mathbf{x}; \boldsymbol{\gamma}_1^{\text{ind}}, \boldsymbol{\Gamma}_1^{\text{ind}})$ , then the local track density is (using eqn. (3.4.25) ),

$$\begin{aligned} P(\mathbf{x}|\mathbf{z}_1) &= P(\mathbf{x}|\mathbf{z}_{1/2})P(\mathbf{x}|\mathbf{z}_1 \cap \mathbf{z}_2) \\ &\propto \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_1, \boldsymbol{\Gamma}_1) \end{aligned} \quad (3.4.27)$$

where

$$(\boldsymbol{\Gamma}_1)^{-1} = \left( \boldsymbol{\Gamma}_m^{-1} + (\boldsymbol{\Gamma}_1^{\text{ind}})^{-1} \right) \quad (3.4.28)$$

$$\hat{\mathbf{x}}_1 = \boldsymbol{\Gamma}_1 \left( (\boldsymbol{\Gamma}_1^{\text{ind}})^{-1} \boldsymbol{\gamma}_1^{\text{ind}} + \boldsymbol{\Gamma}_m^{-1} \boldsymbol{\gamma}_m \right) \quad (3.4.29)$$

which are the local track density parameters for Gaussian case. Similarly for the node  $j$ ,

$$(\boldsymbol{\Gamma}_2)^{-1} = \left( \boldsymbol{\Gamma}_m^{-1} + (\boldsymbol{\Gamma}_2^{\text{ind}})^{-1} \right) \quad (3.4.30)$$

$$\hat{\mathbf{x}}_j = \boldsymbol{\Gamma}_2 \left( (\boldsymbol{\Gamma}_2^{\text{ind}})^{-1} \boldsymbol{\gamma}_2^{\text{ind}} + \boldsymbol{\Gamma}_m^{-1} \boldsymbol{\gamma}_m \right) \quad (3.4.31)$$

If the mutual information component  $\boldsymbol{\Gamma}_m, \boldsymbol{\gamma}_m$  is known, then the optimal fusion is simply the ratio of the product of local densities and the mutual information, as in

eqn. (4.3.3). The result is,

$$\mathbf{\Gamma}^f = ((\mathbf{\Gamma}_1)^{-1} + (\mathbf{\Gamma}_2)^{-1} - \mathbf{\Gamma}_m^{-1}) \quad (3.4.32a)$$

$$\hat{\mathbf{x}}^f = ((\mathbf{\Gamma}_1)^{-1} \hat{\mathbf{x}}_1 + (\mathbf{\Gamma}_2)^{-1} \hat{\mathbf{x}}_2 - \mathbf{\Gamma}_m^{-1} \boldsymbol{\gamma}_m) \quad (3.4.32b)$$

where  $\mathbf{\Gamma}^f, \hat{\mathbf{x}}^f$  are the parameter of fused (Gaussian) density.

It is easy to prove that the resulting cross-correlation matrix according to the proposed correlation structure is,

$$\mathbb{E} \left[ \tilde{\mathbf{x}}_1 (\tilde{\mathbf{x}}_2)^T \right] = \mathbf{\Gamma}_1 \mathbf{\Gamma}_m^{-1} \mathbf{\Gamma}_2 \quad (3.4.33)$$

### 3.4.2 Contradictions in the design of ICI

There are minor flaws in the design of ICI which are also evident from the aforementioned correlation structure it uses. In relation to eqn. (3.4.32), the mutual information component ICI uses are,

$$\mathbf{\Gamma}_m^{\text{ICI}} = (\omega_1 \mathbf{\Gamma}_1 + \omega_2 \mathbf{\Gamma}_2); \quad \boldsymbol{\gamma}_m^{\text{ICI}} = (\omega_1 \hat{\mathbf{x}}_1 + \omega_2 \hat{\mathbf{x}}_2) \quad (3.4.34)$$

which do not form a consistent pair for the estimate of  $\mathbf{x}$  as a mutual component like in the case of HMD-GA. In relation to the correlation structure presented below, this flaw is akin to using a Kalman filter with optimistic sensor/process noise covariance.

In such a case, the bound derived in eqn. (3.4.15) is no longer valid, and inconsistency of  $(\boldsymbol{\gamma}_m^{\text{ICI}}, \mathbf{\Gamma}_m^{\text{ICI}})$  needs to be additionally accounted for. Thus, in case of ICI,

eqn. (3.4.11) becomes,

$$\begin{aligned}
& \left( \omega_2 \mathbf{\Gamma}_1^{-1} \mathbf{\Gamma}_{12} \mathbf{\Gamma}_m^{\text{ICI}^{-1}} + \omega_1 \mathbf{\Gamma}_m^{\text{ICI}^{-1}} \mathbf{\Gamma}_{12} \mathbf{\Gamma}_2^{-1} - \mathbf{\Gamma}_1^{-1} \mathbf{\Gamma}_{12} \mathbf{\Gamma}_2^{-1} \right) + \\
& \left( \omega_1 \mathbf{\Gamma}_2^{-1} \mathbf{\Gamma}_{21} \mathbf{\Gamma}_m^{\text{ICI}^{-1}} + \omega_2 \mathbf{\Gamma}_m^{\text{ICI}^{-1}} \mathbf{\Gamma}_{21} \mathbf{\Gamma}_1^{-1} - \mathbf{\Gamma}_2^{-1} \mathbf{\Gamma}_{21} \mathbf{\Gamma}_1^{-1} \right) \\
& - \mathbf{\Gamma}_m^{\text{ICI}^{-1}} \tilde{\mathbf{\Gamma}} \mathbf{\Gamma}_m^{\text{ICI}^{-1}} \preceq \mathbf{0}
\end{aligned} \tag{3.4.35}$$

Where  $\tilde{\mathbf{\Gamma}}$  is the spread-of-means term since  $\mathbb{E} \left[ \tilde{\boldsymbol{\gamma}}_m^{\text{ICI}} (\tilde{\boldsymbol{\gamma}}_m^{\text{ICI}})^T \right] = \mathbf{\Gamma}_m^{\text{ICI}} + \tilde{\mathbf{\Gamma}}$ . Eqn. (3.4.35) reveals a tighter consistency bound on  $\mathbf{\Gamma}_m^{\text{ICI}^{-1}}$  than in the case of HMD-GA.

In [74], the consistency of ICI was proved by comparing it with optimal track-fusion covariance instead of its own mean-squared error. It should be noted that optimal track-fusion algorithm is only a local optima, and only centralized track fusion is known to be efficient (reaches Cramér-Rao bound [13]), for linear case and full communication rate [12].

The ICI was designed to satisfy the aforementioned correlation structure [74]. With respect to eqn. (3.4.29) and eqn. (3.4.31), the necessary conditions on the covariances involved are,

- As  $\mathbf{\Gamma}_1^{\text{ind}}$  and  $\mathbf{\Gamma}_2^{\text{ind}}$  are the covariance of independent component of  $\mathbf{x}$ , they have to be at least positive semi-definite.
- Since, according to the correlation structure,  $\mathbf{\Gamma}^1$  and  $\mathbf{\Gamma}^2$  are the result of naive fusion of dependent and independent components, the resulting covariance should be smaller than or equal to those components. Thus,

$$\mathbf{\Gamma}_m \geq \mathbf{\Gamma}_1; \quad \mathbf{\Gamma}_m \geq \mathbf{\Gamma}_2 \tag{3.4.36}$$

Now consider the approximation of  $\mathbf{\Gamma}_m$  employed by ICI in eqn. (3.4.34) which suggest a clear violation of eqn. (3.4.36), since  $\mathbf{\Gamma}_m^{\text{ICI}} \geq \min(\mathbf{\Gamma}_1, \mathbf{\Gamma}_2)$  always hold.

For  $\gamma_m = (\omega_1 \hat{\mathbf{x}}_1 + \omega_2 \hat{\mathbf{x}}_2)$ , it can be proved that the requirement for consistency of overall fused estimate is,

$$\mathbf{\Gamma}_m \leq \omega \mathbf{\Gamma}_1 + (1 - \omega) \mathbf{\Gamma}_2 \quad (3.4.37)$$

ICI uses this exact upper bound for the correlation component due to which it is always consistent. But eqns. (3.4.37) and (3.4.36) are in direct contradiction of each other. Therefore, ICI tends to break the exact correlation structure it was derived from.

### 3.4.3 Extending Inverse Covariance Intersection to Gaussian Mixtures

Nevertheless, ICI is a very effective and consistent fusion methodology, which was proposed only for Gaussian densities. Using the fact that ICI can be generated as a special case of HMD-GA by neglecting the spread-of-means terms (and interchanging fusion weights), it is possible to extend it to Gaussian mixtures. Considering the following Gaussian mixtures corresponding to local tracks  $i$  and  $j$ .

$$p(\mathbf{x}|\mathbf{z}^i) = \sum_{m=1}^M \alpha_m \mathcal{N}(\mathbf{x}, \hat{\mathbf{x}}_m, \mathbf{\Gamma}_m), \quad (3.4.38a)$$

$$p(\mathbf{x}|\mathbf{z}^j) = \sum_{n=1}^N \beta_n \mathcal{N}(\mathbf{x}, \hat{\lambda}_n, \mathbf{\Lambda}_n), \quad (3.4.38b)$$

where the notations are self-explanatory. The resulting fused Gaussian mixture using ICI is,

$$p(\mathbf{x}|\mathbf{z}^i \cup \mathbf{z}^j) = \sum_{m=1}^M \sum_{n=1}^N \kappa_{m,n} \alpha_m \beta_n \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{m,n}^f, \mathbf{\Gamma}_{m,n}^f) \quad (3.4.39)$$

where the quantities are,

$$\kappa_{m,n} = \frac{\mathcal{N}(\hat{\mathbf{x}}_m; \hat{\lambda}_n, \mathbf{\Gamma}_m + \mathbf{\Lambda}_n)}{\mathcal{N}(\boldsymbol{\gamma}^{\text{ICI}}; \hat{\mathbf{x}}_{m,n}^{\text{naive}}, \mathbf{\Gamma}^{\text{ICI}} - \mathbf{\Gamma}_{m,n}^{\text{naive}})} \times \frac{|\mathbf{\Gamma}^{\text{ICI}} - \mathbf{\Gamma}_{m,n}^{\text{naive}}|}{|\mathbf{\Gamma}_{m,n}^{\text{naive}}|} \quad (3.4.40)$$

The quantities  $(\hat{\mathbf{x}}_{m,n}^{\text{naive}}, \mathbf{\Gamma}_{m,n}^{\text{naive}})$  refer to the naive fusion of  $m^{\text{th}}$  and  $n^{\text{th}}$  components respectively.

$$\mathbf{\Gamma}_{m,n}^{\text{naive}} = [\mathbf{\Gamma}_m^{-1} + \mathbf{\Lambda}_n^{-1}]^{-1}; \quad \hat{\mathbf{x}}_{m,n}^{\text{naive}} = \mathbf{\Gamma}_{m,n}^{\text{naive}} [\mathbf{\Gamma}_m^{-1} \hat{\mathbf{x}}_m + \mathbf{\Lambda}_n^{-1} \hat{\lambda}_n]$$

The quantities  $(\boldsymbol{\gamma}^{\text{ICI}}, \mathbf{\Gamma}^{\text{ICI}})$  refer to the mutual information component for Gaussian mixtures. These quantities are given by,

$$\boldsymbol{\gamma}^{\text{ICI}} = \omega_1 \sum_{m=1}^M \alpha_m \hat{\mathbf{x}}_m + \omega_2 \sum_{n=1}^N \beta_n \hat{\lambda}_n, \quad \mathbf{\Gamma}^{\text{ICI}} = \omega_1 \sum_{m=1}^M \alpha_m \mathbf{\Gamma}_m + \omega_2 \sum_{n=1}^N \beta_n \mathbf{\Lambda}_n \quad (3.4.41)$$

The final fused mean and covariance for each component in eqn. (3.4.39) are,

$$\mathbf{\Gamma}_{m,n}^f = [\mathbf{\Gamma}_m^{-1} + \mathbf{\Gamma}_n^{-1} - (\mathbf{\Gamma}^{\text{ICI}})^{-1}]^{-1} \quad (3.4.42)$$

$$\hat{\mathbf{x}}_{m,n}^f = \mathbf{\Gamma}_{m,n}^f [\mathbf{\Gamma}_m^{-1} \hat{\mathbf{x}}_m + \mathbf{\Gamma}_n^{-1} \hat{\lambda}_n - (\mathbf{\Gamma}^{\text{ICI}})^{-1} \boldsymbol{\gamma}^{\text{ICI}}] \quad (3.4.43)$$

The result is very similar to the Gaussian mixture implementation of HMD-GA in [84].

### 3.5 Generalized Mean Density Fusion using Importance Sampling.

Unlike HMD, geometric mean density fusion doesn't enjoy applicability to Gaussian mixtures with just one approximation. Although generalized Chernoff fusion and sigma-point-based Chernoff fusion are viable alternatives to the fusion of Gaussian mixtures, sampling-based alternatives are needed in some situations, especially in the case of low-precision systems where matrix inversion is not accurate. The non-positive definiteness of fused covariance matrices is another problem in such scenarios. Sampling methods also provide us with the option of fusing Gaussian and non-Gaussian local tracks together with no approximations. Such methods would also be useful in cases where track densities have to undergo non-linear transformations, like in heterogeneous fusion.

In [3], the concern of Gaussian mixture-based fusion was discussed using importance sampling. The focus of the paper was on different choices of proposal density to sample from. Here we show that due to the nature of expressions of generalized mean density-based fusion, the requirement of proposal density is nullified.

We explain the procedure of calculating a generalized expectation of a function using importance sampling in the case of GMD by approximating its expression with a Gaussian. This requires calculating its first- and second-order moments. Using importance sampling, the expectation of a function  $f(\mathbf{x})$  with respect to a density  $p(\mathbf{x})$  can be calculated as

$$\mathbb{E}[f(\mathbf{x})]_{p(\mathbf{x})} = \int f(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} = \int \frac{f(\mathbf{x}) p(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x}) d\mathbf{x} = \mathbb{E} \left[ \frac{p(\mathbf{x})}{q(\mathbf{x})} f(\mathbf{x}) \right]_{q(\mathbf{x})} \quad (3.5.1)$$

where  $q(\mathbf{x})$  is the proposal density which is relatively easy to sample from, in comparison to  $p(\mathbf{x})$ . The expectation can then be computed using samples  $\mathbf{x}_n$  from  $q(\mathbf{x})$  as,

$$\mathbb{E} \left[ \frac{p(\mathbf{x})}{q(\mathbf{x})} f(\mathbf{x}) \right]_{q(\mathbf{x})} = \frac{1}{N} \sum_{n=1}^N \frac{f(\mathbf{x}_n) p(\mathbf{x}_n)}{q(\mathbf{x}_n)} \quad (3.5.2)$$

where  $N$  is the number of samples.

Now, Lets observe the expression for geometric mean density fusion,

$$p^g(\mathbf{x}) \propto \frac{p^1(\mathbf{x})p^2(\mathbf{x})}{(p^1(\mathbf{x}))^{\omega_2} (p^2(\mathbf{x}))^{\omega_1}} \quad (3.5.3)$$

where  $p^1(\mathbf{x}), p^2(\mathbf{x})$  are the local track densities and  $\omega_1, \omega_2$  are the fusion weights. The proportionality constant is the normalization factor  $\zeta^g$  such that the  $p^g(\mathbf{x})$  is a valid density. For now, let's assume that  $p^1(\mathbf{x})$  is Gaussian (or easy to sample from).

Due to the structure of GMD, a generalized expectation with respect to  $p^g(\mathbf{x})$  can be expressed as,

$$\begin{aligned} \mathbb{E}[f(\mathbf{x})]_{p^g(\mathbf{x})} &= \frac{1}{\zeta^g} \int f(\mathbf{x}) \frac{p^1(\mathbf{x})p^2(\mathbf{x})}{(p^1(\mathbf{x}))^{\omega_2} (p^2(\mathbf{x}))^{\omega_1}} d\mathbf{x} \\ &= \frac{1}{\zeta^g} \int f(\mathbf{x}) \frac{p^2(\mathbf{x})}{(p^1(\mathbf{x}))^{\omega_2} (p^2(\mathbf{x}))^{\omega_1}} p^1(\mathbf{x}) d\mathbf{x} \\ &= \mathbb{E} \left[ f(\mathbf{x}) \frac{p^2(\mathbf{x})}{(p^1(\mathbf{x}))^{\omega_2} (p^2(\mathbf{x}))^{\omega_1}} \right]_{p^1(\mathbf{x})} \end{aligned} \quad (3.5.4)$$

Due to Gaussianity of  $p^1(\mathbf{x})$ , it is easy to sample from. The resulting expectation can



be computed using the following summation,

$$\mathbb{E}[f(\mathbf{x})]_{p^g(\mathbf{x})} = \frac{1}{\zeta^g} \sum_{n=1}^N f(\mathbf{x}_n) \frac{p^2(\mathbf{x}_n)}{(p^1(\mathbf{x}_n))^{\omega_2} (p^2(\mathbf{x}_n))^{\omega_1}} \quad (3.5.5)$$

with  $\mathbf{x}_n$  being samples from  $p^1(\mathbf{x})$ . Note that due to symmetry of the mean-density expressions, it is also possible to obtain the expectation in question using samples from  $p^2(\mathbf{x})$

$$\mathbb{E}[f(\mathbf{x})]_{p^g(\mathbf{x})} = \mathbb{E} \left[ f(\mathbf{x}) \frac{p^1(\mathbf{x})}{(p^1(\mathbf{x}))^{\omega_2} (p^2(\mathbf{x}))^{\omega_1}} \right]_{p^2(\mathbf{x})} \quad (3.5.6)$$

In comparison to HMD, the expression for GMD in eqn. (3.5.3) only differs in the approximation of mutual information component in the denominator. Hence HMD also enjoys sampling based evaluation with some obvious changes in eqns. (3.5.4) and (3.5.6). Note that the normalization constant also needs to be calculated which is given by,

$$\zeta^g = \mathbb{E}[1] = \sum_{n=1}^N \frac{p^1(\mathbf{x}_n)}{(p^1(\mathbf{x}_n))^{\omega_2} (p^2(\mathbf{x}_n))^{\omega_1}} = \sum_{n=1}^N \frac{p^2(\mathbf{x}_n)}{(p^1(\mathbf{x}_n))^{\omega_2} (p^2(\mathbf{x}_n))^{\omega_1}} \quad (3.5.7)$$

where, again, samples from either  $p^1(\mathbf{x}_n)$  or  $p^2(\mathbf{x}_n)$  can be used with relevant expression. The evaluation of normalization constant for HMD follows similarly.

### 3.5.1 Sampling based fusion of Gaussian Mixtures

Considering the Gaussian mixtures in eqn. (3.4.38), the resulting GMD is then,

$$p^g(\mathbf{x}) = \sum_{m=1}^M \sum_{n=1}^N \alpha_m \beta_n \frac{\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_m, \mathbf{\Gamma}_m) \mathcal{N}(\mathbf{x}; \hat{\lambda}_n, \mathbf{\Lambda}_n)}{(p(\mathbf{x}|\mathbf{z}^i))^{\omega_2} (p(\mathbf{x}|\mathbf{z}^j))^{\omega_1}} \quad (3.5.8)$$

The trick we are going to use here is to approximate the last fraction in the R.H.S. of eqn. (3.5.8) as a Gaussian density using the sampling method proposed before.

Thus,

$$\frac{\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_m, \mathbf{\Gamma}_m) \mathcal{N}(\mathbf{x}; \hat{\lambda}_n, \mathbf{\Lambda}_n)}{(p(\mathbf{x}|\mathbf{z}^i))^{\omega_2} (p(\mathbf{x}|\mathbf{z}^j))^{\omega_1}} \approx \frac{1}{\zeta} \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_s^{m,n}, \mathbf{\Gamma}_s^{m,n}) \quad (3.5.9)$$

where  $\zeta$  is the scaling factor so that the ratio is a valid probability distribution. The mean,  $\hat{\mathbf{x}}_s^{m,n}$  is given by,

$$\begin{aligned} \hat{\mathbf{x}}_s^{m,n} &= \frac{1}{\zeta} \sum_{s=1}^S \mathbf{x}_s \frac{\mathcal{N}(\mathbf{x}_s; \hat{\lambda}_n, \mathbf{\Lambda}_n)}{(p(\mathbf{x}_s|\mathbf{z}^i))^{\omega_2} (p(\mathbf{x}_s|\mathbf{z}^j))^{\omega_1}} \\ &= \frac{1}{\zeta} \sum_{s=1}^S \mathbf{x}_s \frac{\mathcal{N}(\mathbf{x}_s; \hat{\mathbf{x}}_m, \mathbf{\Gamma}_m)}{(p(\mathbf{x}_s|\mathbf{z}^i))^{\omega_2} (p(\mathbf{x}_s|\mathbf{z}^j))^{\omega_1}} \end{aligned} \quad (3.5.10)$$

depending upon whether  $\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_m, \mathbf{\Gamma}_m)$  or  $\mathcal{N}(\mathbf{x}_s; \hat{\lambda}_n, \mathbf{\Lambda}_n)$  is used for sampling. Calculation of  $\mathbf{\Gamma}_s^{m,n}$  follows similarly.

$$\mathbf{\Gamma}_s^{m,n} = \frac{1}{\zeta} \sum_{s=1}^S (\mathbf{x}_s)(\mathbf{x}_s)^T \frac{\mathcal{N}(\mathbf{x}_s; \hat{\lambda}_n, \mathbf{\Lambda}_n)}{(p(\mathbf{x}_s|\mathbf{z}^i))^{\omega_2} (p(\mathbf{x}_s|\mathbf{z}^j))^{\omega_1}} - (\hat{\mathbf{x}}_s^{m,n})(\hat{\mathbf{x}}_s^{m,n})^T, \quad (3.5.11)$$

when  $\mathcal{N}(\mathbf{x}_s; \hat{\mathbf{x}}_m, \mathbf{\Gamma}_m)$  is used for sampling. Note that calculation of  $\zeta$  is required which is straightforward as in eqn. (3.5.7). The resulting fused Gaussian mixture is then,

$$p^g(\mathbf{x}) = \sum_{m=1}^M \sum_{n=1}^N \alpha_m \beta_n \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_s^{m,n}, \mathbf{\Gamma}_s^{m,n}) \quad (3.5.12)$$

Similar results can be derived for HMD.

### 3.5.2 Which Density to Sample From ?

An intuition for fused density is that it lies at the intersection of local track densities. Depending on the number of samples, sampling from a single distribution might lead the fused density to orient towards the only distribution sampled from. Thus, if the problem enjoys the flexibility of sampling from both distributions, one can opt to utilize samples from one distribution over the other. This depends on the normalization weights in the importance sampling method.

Considering eqns. (3.5.4) and (3.5.6), suppose the user samples  $\mathbf{x}_s^1$  and  $\mathbf{x}_s^2$  from pdfs  $p^1(\mathbf{x})$  and  $p^2(\mathbf{x})$  respectively. Then, one should choose the sample which gives a larger value of normalization weight. Therefore, choose  $\mathbf{x}_s^1$  if,

$$\frac{p^2(\mathbf{x}_s^1)}{(p^1(\mathbf{x}_s^1))^{\omega_2} (p^2(\mathbf{x}_s^1))^{\omega_1}} \geq \frac{p^1(\mathbf{x}_s^2)}{(p^1(\mathbf{x}_s^2))^{\omega_2} (p^2(\mathbf{x}_s^2))^{\omega_1}} \quad (3.5.13)$$

else, choose  $\mathbf{x}_s^2$ . Note that in contemporary systems, sampling from normal distribution can be easily parallelized, resulting in much faster processing using a GPU.

Finally, for demonstrating the case of fusing two Gaussian densities using the proposed sampling strategy, an example is set up, the results of which are shown in

Fig. 3.3. The local track densities are,

$$p^1(\mathbf{x}) = \mathcal{N}\left(\mathbf{x}; \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}, \begin{bmatrix} 2.5 & -1 \\ -1 & 1.2 \end{bmatrix}\right), \quad p^2(\mathbf{x}) = \mathcal{N}\left(\mathbf{x}; \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 0.8 & -0.5 \\ -0.5 & 4 \end{bmatrix}\right) \quad (3.5.14)$$

where 5000 samples were used from both the distribution based on eqn. (3.5.13). As expected, the Geometric mean-density fusion, which is an analogue of CI, has the largest uncertainty region. Note that covariance of HMD-GA is less than HMD-S which is obvious since the mutual information component in the former is approximated by a Gaussian distribution which has the largest entropy and hence the highest covariance. Therefore, the resulting fused covariance is minimum.

The sampling method allow us to vary conservativeness with ease. To demonstrate this, we use an inflation factor  $\alpha \geq 1$ , and sample from the same densities but inflated covariance,  $\mathbf{\Gamma}' = \alpha\mathbf{\Gamma}$ . The resulting uncertainty regions are shown in Fig. 3.3b. Small variations in  $\alpha$  lead to a resulting covariance even grater than GMD, without any changes in computation. Such results would be helpful in track fusion cases where the local densities suffer from inconsistency, track bias, anomalies or adversarial attacks.

For the case of Gaussian mixtures, we consider the following track densities,

$$p^1(\mathbf{x}) = 0.3\mathcal{N}\left(\mathbf{x}; \begin{bmatrix} -0.5 \\ 3 \end{bmatrix}, \begin{bmatrix} 2.5 & -1 \\ -1 & 1.2 \end{bmatrix}\right) + 0.7\mathcal{N}\left(\mathbf{x}; \begin{bmatrix} 2 \\ 0.3 \end{bmatrix}, \begin{bmatrix} 0.8 & -0.5 \\ -0.5 & 4 \end{bmatrix}\right)$$

$$p^2(\mathbf{x}) = 0.4\mathcal{N}\left(\mathbf{x}; \begin{bmatrix} -1.5 \\ 1 \end{bmatrix}, \begin{bmatrix} 2.5 & -1 \\ -1 & 1.2 \end{bmatrix}\right) + 0.6\mathcal{N}\left(\mathbf{x}; \begin{bmatrix} 3 \\ -4 \end{bmatrix}, \begin{bmatrix} 0.8 & -0.5 \\ -0.5 & 4 \end{bmatrix}\right)$$

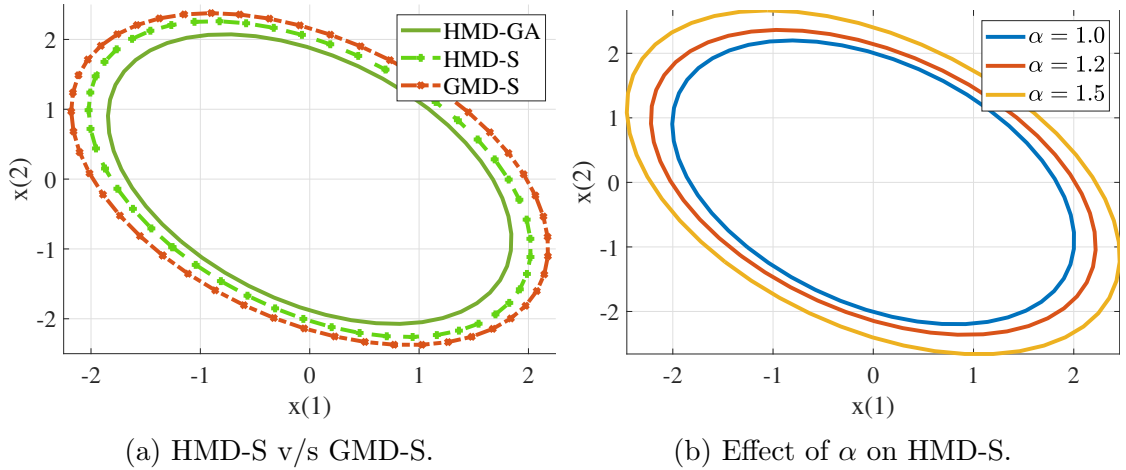


Figure 3.3: Uncertainty region (86.5%) for HMD and GMD fusion of Gaussian densities using proposed sampling technique.

The corresponding mode covariance is kept the same in both the densities without any loss of generalization. The local densities are plotted in Figs. 3.4a and 3.4b, and the corresponding fused mixtures are plotted in Figs. 3.4c and 3.4d for the cases of GMD and HMD, respectively. Interestingly, both the HMD and GMD mixtures lie at the intersection of local uncertainty regions, as in the case of Gaussian densities, with the uncertainty in HMD-S being less than that in GMD-S, as expected. Again, 5000 samples were used in the simulation.

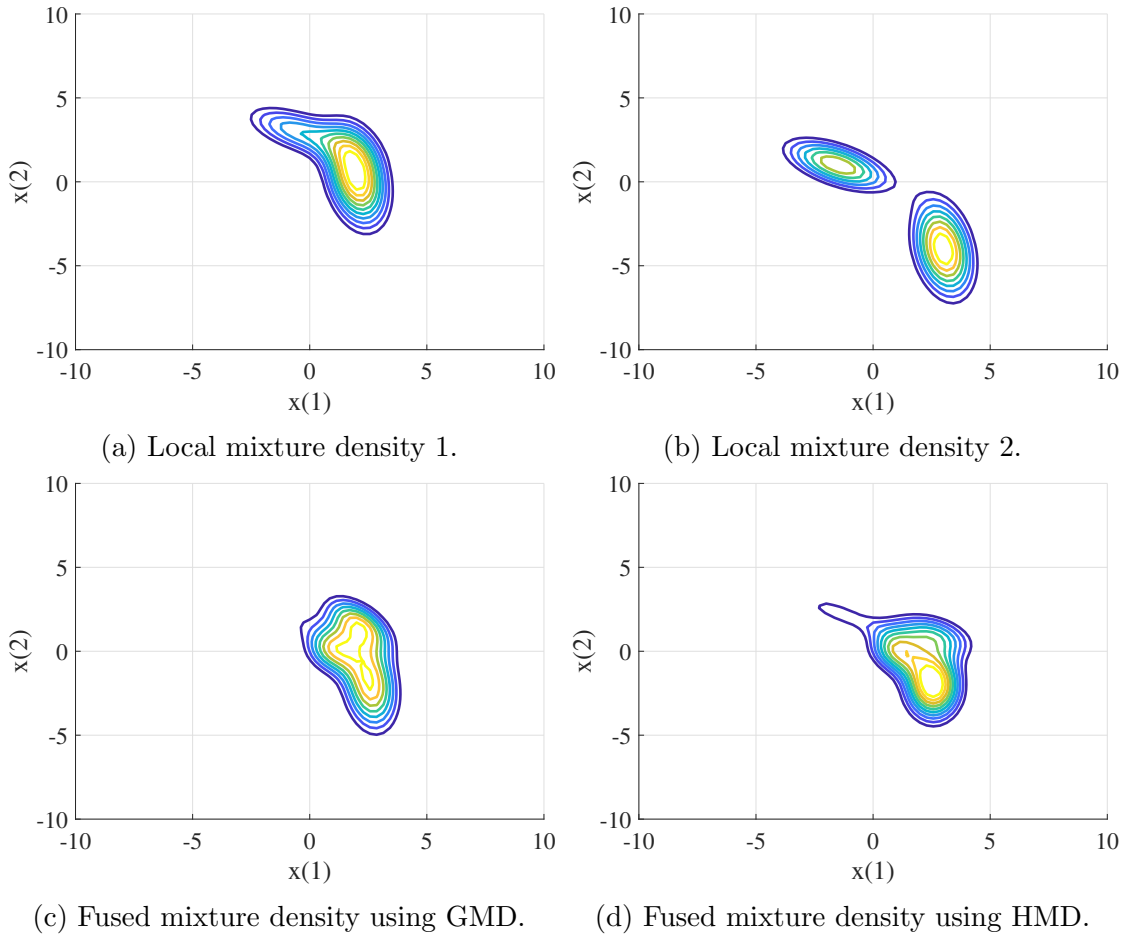


Figure 3.4: Fusion of Gaussian mixtures using proposed sampling technique.

## 3.6 Simulations

To assess the performance of HMD, especially its consistency, we performed a rigorous evaluation based on three different simulation scenarios. Note that HMD-GA was shown to produce the least covariance, which is also shown in Fig. 3.3a. Therefore, we preferred using HMD-GA for comparison in order to establish empirical consistency. In sampling-based methods, the main question of consistency can be tackled easily,

which is evident in Fig. 3.3b, which is why we kept it out of this section. Another reason for not testing sampling-based methods is the computational time involved in sampling in addition to Monte-Carlo (MC) runs.

We first discuss the evaluation of fusion weights for the proposed HMD-GA fusion. The findings are interesting in the case of scalar estimates. Among the simulation scenarios, the first two cases are aimed at only consistency, and no error-based performance metrics are employed. The third simulation is aimed at evaluating the performance of HMD-GA in a real-life multi-target, multi-sensor tracking scenario wherein the fusion center is equipped with memory. Due to the huge sensor coverage of 300 km, the local tracks become inconsistent due to the curvature of cross-range uncertainty, and thus the consistency of the fusion methodology cannot be guaranteed among any of the fusion methodologies discussed in this chapter.

The aim of this section is to deem HMD, especially HMD-GA, a viable candidate for generalized fusion of processed densities, which is divided into two phases based on the aim, viz., consistency-based and performance-based scenarios.

### 3.6.1 Fusion Weights

The fundamental basis of computing fusion weight  $\omega$  in all conservative fusion techniques is to minimize some scalar function of output (reported) covariance. Throughout the literature, this problem has been perceived from the angles of minimizing the entropy of the fused estimate, or in Mahler's words, to increase the peakiness of the fused distribution [49], both of which tend to incline towards minimizing the fused covariance in the Gaussian case.

Based on such heuristics, we try to optimize  $\omega$  such that output covariance is

minimized, but from a different angle. We attempt to maximize the mutual information component, which also results in the minimization of fused covariance. The optimization becomes,

$$\omega^* = \arg \min_{\omega} f \left[ \left( (1 - \omega)\mathbf{\Gamma}_1 + \omega\mathbf{\Gamma}_2 + \tilde{\mathbf{\Gamma}} \right)^{-1} \right] \quad (3.6.1)$$

where  $\tilde{\mathbf{\Gamma}}$  which also depends on  $\omega$ , is defined in eqn. (3.4.4); and  $f[\cdot]$  is any scalar representation of a matrix like a determinant or trace. The reason for a different optimization approach is that covariance inflation is not a problem in HMD-GA, so the user should not be concerned about minimizing it. Note that this optimization is slightly faster to compute as compared to that in CI and ICI since it requires only one matrix inversion compared to 2 and 3 in CI and ICI, respectively.

Table 3.1 reports the computation times for implementing different fusion algorithms relative to naive fusion. All algorithms calculate fusion weights to minimize their respective covariances whereas eqn. (3.6.1) is employed in the case of HMD-GA. Ellipsoidal intersection (EI) [85], which does not require calculating fusion weights, is also included for comparison.

Table 3.1: Relative computational time of fusion algorithms.

	Naive	CI	ICI	EI	HMD-GA
<b>Relative Time</b>	1	3.74	4.43	4.09	3.68

An interesting point can be observed in the case of scalar estimates. The fusion



weight in the case of CI is calculated as ( $\Gamma_1, \Gamma_2$  are scalars),

$$\begin{aligned}
 \omega^* &= \arg \min_{\omega} [\omega \Gamma_1^{-1} + (1 - \omega) \Gamma_2^{-1}]^{-1} \\
 &= \arg \min_{\omega} [\Gamma_1 + \omega (\Gamma_2 - \Gamma_1)]^{-1} \\
 &= \begin{cases} 0, & \text{if } \Gamma_1 > \Gamma_2 \\ 1, & \text{if } \Gamma_2 > \Gamma_1 \end{cases} \tag{3.6.2}
 \end{aligned}$$

Similarly in the case of ICI, the given optimization problem becomes,

$$\begin{aligned}
 \omega^* &= \arg \min_{\omega} \left[ \frac{1}{\Gamma_1} + \frac{1}{\Gamma_2} - \frac{1}{\omega \Gamma_1 + (1 - \omega) \Gamma_2} \right]^{-1} \\
 &= \arg \max_{\omega} [\omega \Gamma_1 + (1 - \omega) \Gamma_2] \\
 &= \begin{cases} 1, & \text{if } \Gamma_1 > \Gamma_2 \\ 0, & \text{if } \Gamma_2 > \Gamma_1 \end{cases} \tag{3.6.3}
 \end{aligned}$$

Thus, the result in both cases is the local track estimate corresponding to the least covariance. This property renders using CI and ICI useless in the case of scalar track fusion, even though the results are consistent (assuming participating densities are consistent). Due to the presence of the spread-of-means term, HMD-GA does not suffer from any such problem.

To support this claim, a scalar parameter estimation problem was set up wherein

the state  $x_k$  and measurements  $z_k$  at instant  $k$  are described by the following equations,

$$x_k = x_{k-1} + w_{k-1} \quad (3.6.4)$$

$$z_k = x_k + v_k \quad (3.6.5)$$

where the noise parameters  $w_{k-1}$  and  $v_k$  are zero-mean Gaussian and mutually correlated with a correlation coefficient  $\rho$ . The correlation is assumed to be unknown, and at each instant, the prior distribution is attempted to fuse with the measurement distribution using conservative techniques. The fusion weight  $\omega$ , averaged over 500 Monte-Carlo runs, is shown in Fig. 3.5. As observed, the fusion weight in the case of CI is always 0, and in the case of ICI, it is always 1. Thus, the output of these algorithms is the measurement itself. The figure proves that the only fusion performed here is in the case of HMD-GA, with an average fusion weight value 0.2. Thus, measurement is given more weight, but prior information is also fused.

### 3.6.2 Consistency-Based Test 1

For consistency based test, we re-developed the scenarios employed in [73] and [74]. The tests are aimed at testing consistency by visualizing covariance ellipsoids after averaging them over a large number of Monte-Carlo runs. The setup for the first test consists of 10 nodes in a configuration as shown in Fig. 3.6a. The transmission of information follows the direction of arrows in the figure. For e.g. node S7 receives estimates from nodes S4 and S5 which also gather estimates from nodes S1 and S2 respectively. Network flow for rest of the nodes follow similarly. Each node is

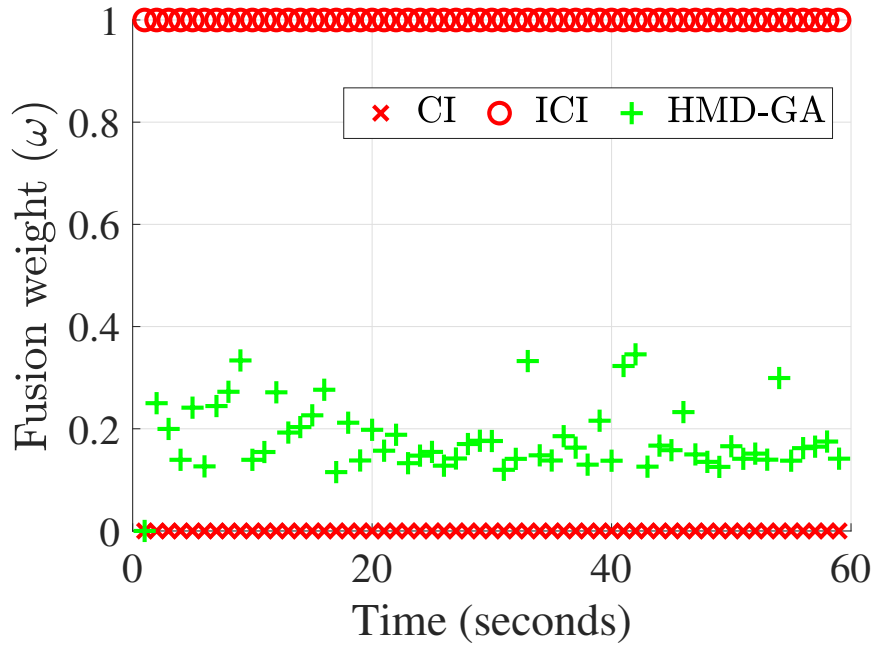


Figure 3.5: Comparison of fusion weights in case of scalar estimates.

capable of processing the local measurements using a local Kalman filter (KF) before transmitting or fusing it with incoming tracks as per the network. No ambiguity on time-synchronization or sensor-bias has been included in the problem.

During an MC run, each node is initialized by sampling from the distribution,  $\mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}\right)$ . The measurement matrix at each node  $i$  is,

$$\mathbf{H}_i = \begin{bmatrix} \sin\left(\frac{\pi}{2} \cdot \frac{i}{10}\right) & \cos\left(\frac{\pi}{2} \cdot \frac{i}{10}\right) \\ \cos\left(\frac{\pi}{2} \cdot \frac{i}{10}\right) & \sin\left(\frac{\pi}{2} \cdot \frac{i}{10}\right) \end{bmatrix}, \quad (3.6.6)$$

in addition to the zero-mean Gaussian measurement noise with variance  $R_i = 0.2\mathbf{I}_2$ . Initial estimate at each node is updated using the local measurement using a standard KF and then fused with other nodes along the path. At the end of transmission,

fused estimates are received at node S10, where we compare the reported covariance ellipsoids for each fusion methodology with the sample covariance ellipsoids, averaged over 50,000 MC runs. The algorithms used for comparison are covariance intersection, inverse-covariance intersection, naive fusion, among which the latter is proven to be inconsistent. Also, as a baseline, centralized fusion is employed which is optimal in the sense of minimum-mean square error.

Among the covariance ellipsoids plotted in Fig. 3.6a, it can be seen that centralized fusion is consistent with respect to the sample covariance,  $\mathbb{E} [(\hat{\mathbf{x}}_i - \mathbf{x})(\hat{\mathbf{x}}_i - \mathbf{x})^T]$ . The naive fusion produces a highly inconsistent ellipsoid as the reported covariance overestimates the actual error-covariance by a big margin. This is obviously due to double counting of common information. As expected, the ICI and CI produce conservative estimates with CI producing a heavily bloated covariance compared to the sample covariance. Not surprisingly, the least conservative result is produced by HMD-GA with actual error even tighter than ICI.

### 3.6.3 Consistency-Based Test 2

The source of correlation in the first scenario was only due to process noise, since there was no measurement history. To check the performance of the proposed fusion algorithm due to the presence of both correlations, we evaluated the consistency of HMD-GA on another scenario presented in Fig. 3.6b. The difference here is that the local nodes are allowed to process measurements over five time steps before the estimates are sent to node S5, as per the network configuration. Again, the reported covariance is compared with the sampled covariance averaged over 50,000 MC runs. Since the two-dimensional state is now evolving with time, the following linear model

is used for propagation,

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{w}_k \quad (3.6.7)$$

where the state-transition matrix  $\mathbf{F}_k = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix}$ , the zero-mean process noise  $\mathbf{w}_k$  follows the distribution  $\mathcal{N}(\mathbf{0}, 0.5\mathbf{I}_2)$  with  $\mathbf{I}_n$  being an n-dimensional identity matrix. The observation model at node  $i$  is,

$$\mathbf{z}_k^i = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k^i \quad (3.6.8)$$

where  $\mathbf{H}_k = \mathbf{I}_2$  and  $\mathbf{v}_k^i$  is sampled from a zero-mean Gaussian distribution with covariance  $\begin{bmatrix} 0.5 & 0 \\ 0 & 0.2 \end{bmatrix}$  for nodes S1, S3 and S5; and  $\begin{bmatrix} 0.1 & 0 \\ 0 & 0.5 \end{bmatrix}$  for nodes S2, S4. The initial estimate at each node is sampled from the distribution  $\mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\right)$ .

The results are shown in Fig. 3.8, wherein again, the centralized fusion is completely consistent, whereas the naive result severely fails to produce a consistent estimate. In this scenario as well, HMD-GA produces the least conservative result, followed by ICI, and then CI, which produces the largest covariance. Also note that along with the reported covariance, the actual error in the case of HMD-GA is also less compared to other conservative fusion methods, with the size of the covariance ellipsoid almost equaling the centralized fusion result. In both of these cases, HMD-GA has been shown to perform well in scenarios with both sources of correlation.

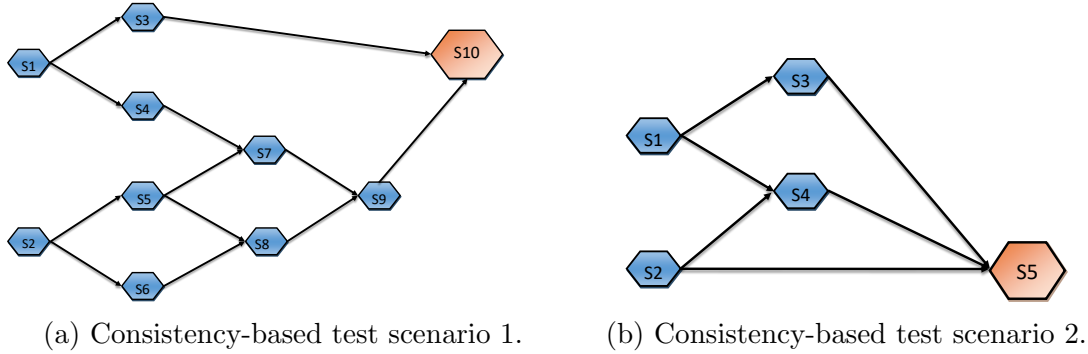


Figure 3.6: Scenarios : Consistency based tests.

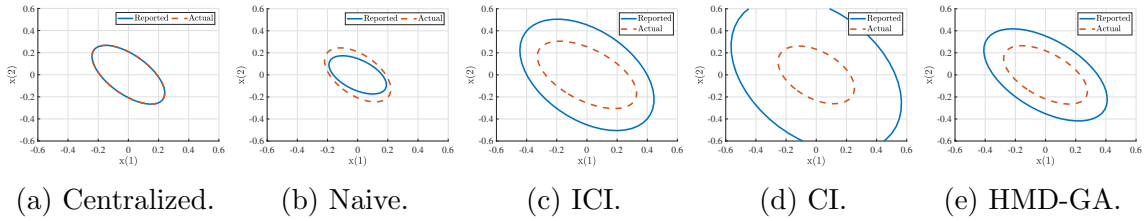


Figure 3.7: Fusion result for consistency-based test 1.

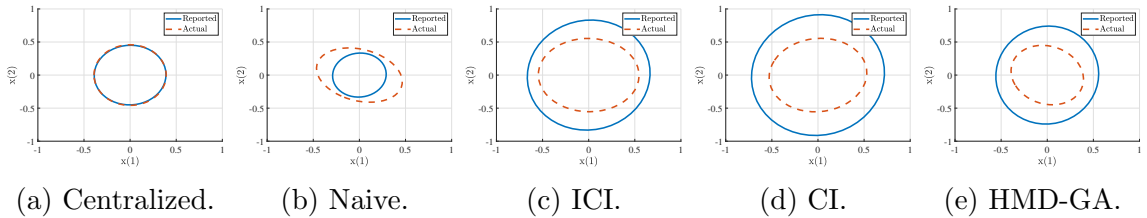


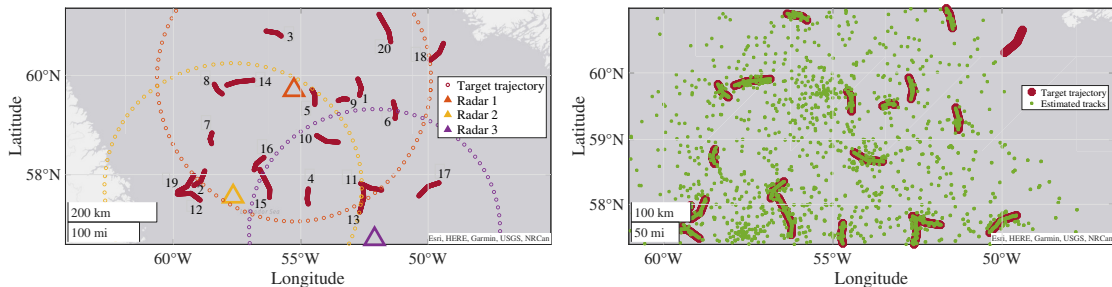
Figure 3.8: Fusion result for consistency-based test 2.

### 3.6.4 A Practical Multi-Target Tracking Scenario.

While, the tests in the previous subsections prove the consistency of the proposed algorithm, they lack practicality in the sense that they are oversimplified. For example, the linear measurement assumption guarantees that the local Kalman filter is consistent. In addition, the total number of measurements was low in comparison to a real-life scenario to prove that the repeated fusion of the same track using the proposed method doesn't hamper consistency. An anticipated metric such as

root-mean-square error (RMSE) is also missing in the earlier simulations.

This scenario consists of three radars surrounded by 20 targets within a significant amount of clutter, as shown in Fig. 3.9a. The sensors have a limited field of view (FOV), as can be seen with the circular markers around the radar in the same figure. All the radars are equipped with local processors such that only updated tracks are sent to the fusion center (FC) every  $T_f$  seconds. The FC is assumed to be collocated with Radar 1. As in the previous simulations, ambiguity in time synchronizations and sensor biases have been ignored in this scenario. The scenario parameters are listed in Table 4.3. The targets move in an area spanning nearly  $3600 \text{ km}^2$  whereas the sensors are static at coordinates mentioned in Table 3.3. Geodetic coordinates of targets according to the WGS84 datum are also provided for replication. Note that a nearly constant velocity (NCV) model is used in tracking, which the target doesn't adhere to, resulting in spikes in the RMSE plots. We tackle this phenomenon by using a relatively high process noise intensity. The scenario is considered sufficiently pragmatic due to the following features :



(a) Multi-target multi-sensor scenario used in section 3.6.4. (b) Finalized tracks (among clutter) using HMD-GA.

- (i) On average, the targets are at least a hundred kilometers far from a sensor. The curvature of cross-range uncertainty is a known phenomenon while tracking at

Table 3.2: Simulation parameters – Scenario 2

Parameter	Value
Process noise intensity ( $\tilde{q}$ )	0.15 m <sup>2</sup> /s <sup>3</sup>
Number of targets	20
Detection probability	0.99
False alarm probability ( $P_{fa}$ )	10 <sup>-6</sup>
Clutter density ( $\lambda$ )	10 <sup>-6</sup>
Maximum target speed	30 m/s
Number of sensors	3
Sensor type	Monostatic 2-D Radar
Sensor coverage	300,000 m × 360°
Sensor std. deviation - range ( $\sigma_r$ )	50 m
Sensor std. deviation - azimuth ( $\sigma_\theta$ )	2°
Gating threshold ( $\mu$ )	0.95
Track deletion threshold (num. of misses)	6
Track loss threshold	500 m
Fusion sample period ( $T_f$ )	10 seconds
Total simulation time	4537 seconds
Sensor sample time	2 seconds

such distances, due to which local consistency is impacted. For more details on this topic, readers are referred to [93].

- (ii) Local trackers do not enjoy a clear view. Due to the high volume of clutter, the probability data association (PDA) tracker [53] is employed.
- (iii) The targets do not necessarily adhere to the state-space motion model employed by local trackers.
- (iv) Track association at the fusion center is performed without knowledge of the correlation between the tracks.
- (v) The target position is captured in geodetic coordinates, which are converted to local east-north-up (ENU) for tracking.



Table 3.3: Locative Parameters – Scenario 2

Parameter	Initial coordinates ( $^{\circ}\text{N}, ^{\circ}\text{E}$ )	Final coordinates ( $^{\circ}\text{N}, ^{\circ}\text{E}$ )
Target 1	59.6104, -52.6939	59.9352, -52.7611
Target 2	57.7699, -59.1922	58.0822, -58.7635
Target 3	60.7648, -55.762	60.8607, -56.3676
Target 4	57.7025, -54.7059	57.3769, -54.7036
Target 5	59.4254, -54.4498	59.7284, -54.5691
Target 6	59.1483, -51.2977	59.4869, -51.3718
Target 7	58.8561, -58.484	58.6288, -58.495
Target 8	59.831, -58.3974	59.6367, -58.0819
Target 9	59.5306, -53.1629	59.5036, -53.5418
Target 10	58.8028, -54.3803	58.6656, -53.5222
Target 11	57.8132, -52.6504	57.676, -51.8147
Target 12	57.4636, -58.9469	57.5906, -59.8504
Target 13	57.2096, -52.6824	57.7325, -52.5499
Target 14	59.9093, -56.8477	59.8108, -57.9123
Target 15	57.5181, -56.2302	58.1104, -56.6123
Target 16	58.3548, -56.4447	58.0948, -56.8321
Target 17	57.8268, -49.5721	57.55, -50.3705
Target 18	60.2969, -49.9242	60.6296, -49.3978
Target 19	57.6451, -59.8337	57.988, -59.1956
Target 20	60.6533, -51.4683	61.1832, -51.9363
Radar 1	59.7138694, -55.2676093	-
Radar 2	57.5399008, -57.6551522	-
Radar 3	56.6320564, -52.104272	-

## Model Equations

Although the coordinates mentioned represent a spherical system, the trackers are implemented in a standard mixed coordinate system, wherein the process model is Cartesian and the sensor model is spherical in two dimensions. For conversion between the geodetic and local tangent plane coordinates, readers are referred to [28].

The target motion and the measurements, respectively, are modeled by the following equations at time  $k$  (dropping target and measurement indices),

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{w}_k, \quad (3.6.9a)$$

$$\mathbf{z}_k = \mathbf{h}(x_k, y_k) + \mathbf{v}_k, \quad (3.6.9b)$$

where (using  $\mathbf{I}_n$  for  $n$  dimensional identity matrix),

$$\mathbf{F} = \begin{bmatrix} \mathbf{I}_2 & \Delta T \mathbf{I}_2 \\ \mathbf{0}_2 & \mathbf{I}_2 \end{bmatrix} \quad \text{and, } \mathbf{h}(x_k, y_k, z_k) = \begin{bmatrix} r_k & \theta_k \end{bmatrix}^T,$$

where,

$$r_k = \sqrt{x_k^2 + y_k^2}; \quad \theta_k = \tan^{-1} \left( \frac{y_k}{x_k} \right)$$

$\Delta T$  is the local sampling time, and  $\mathbf{w}_k$ ,  $\mathbf{v}_k$  are uncorrelated, zero-mean Gaussian distributed process noise vector and measurement noise vector respectively, such that  $\forall \{k, j\}$ ,

$$E[\mathbf{w}_k \mathbf{w}_j] = \delta_{kj} \mathbf{Q}; \quad E[\mathbf{w}_k \mathbf{v}_j] = \mathbf{0}; \quad E[\mathbf{v}_k \mathbf{v}_j] = \delta_{kj} \mathbf{R},$$

where  $\mathbf{Q}$  and  $\mathbf{R}$  are the process noise and measurement noise covariance matrix respectively,

$$\mathbf{Q} = \tilde{q} \begin{bmatrix} \frac{\Delta T^3}{3} \mathbf{I}_2 & \frac{\Delta T^2}{2} \mathbf{I}_2 \\ \frac{\Delta T^2}{2} \mathbf{I}_2 & \Delta T \mathbf{I}_2 \end{bmatrix}, \quad \mathbf{R} = \text{diag}(\sigma_r^2, \sigma_\theta^2),$$

and  $\tilde{q}$  is the process noise intensity in meter<sup>2</sup>/second<sup>3</sup>.

Due to high clutter volume, the local tracking algorithm employs PDA, thereby, indirectly assuming wide separation of targets, which is the case here. The PDA algorithm assigns weights to all available measurement in a single scan, and perform measurement update by taking a weighted average of all innovations. Since the targets are widely separated, there can only be one target-originated measurement in a gate. After local processing, the updated tracks are sent to FC (Radar 1), every  $T_f = 10$  seconds.

The fusion center is equipped with memory. Thus, a global track is always maintained for each fused track at least until the next fusion step. Track-management at FC is similar to that at local node. When a track-list arrives at FC, it is associated with existing global track lists by solving the following 2-D optimization based cost,

$$J = \sum_{i=1}^{N_g} \sum_{j=1}^{N_l} C_{ij} z_{ij}, \quad \text{subjected to constraints,} \quad (3.6.10)$$

$$\sum_{i=1}^{N_g} z_{ij} = 1; \quad \sum_{j=1}^{N_l} z_{ij} = 1, \quad \forall i, j \quad (3.6.11)$$

where  $z_{ij}$  is the assignment between  $i^{\text{th}}$  global track and  $j^{\text{th}}$  local track. The corresponding cost  $C_{ij}$  is defined as,

$$\begin{aligned} C_{ij} &= (\hat{\mathbf{x}}^i - \hat{\mathbf{x}}^j)^T [\mathbf{\Gamma}^i + \mathbf{\Gamma}^j - \mathbf{\Gamma}^{ij} - \mathbf{\Gamma}^{ji}] (\hat{\mathbf{x}}^i - \hat{\mathbf{x}}^j) \\ &\approx (\hat{\mathbf{x}}^i - \hat{\mathbf{x}}^j)^T [\mathbf{\Gamma}^i + \mathbf{\Gamma}^j] (\hat{\mathbf{x}}^i - \hat{\mathbf{x}}^j) \end{aligned} \quad (3.6.12)$$

due to the unavailability of  $\mathbf{\Gamma}^{ij} = (\mathbf{\Gamma}^{ji})^T$ . Next step is the fusion of associated tracks wherein we used ICI, CI and the proposed HMD-GA. This procedure is repeated until track-lists from all local nodes have been fused.

### Performance Metrics

The simulation is evaluated on following metrics :

- (i) **Root mean-square error (RMSE)** : RMSE evaluates the absolute filtering performance in simulation where the true system trajectory is known. At any time-step  $k$ , the RMSE is given by the equation,

$$\text{RMSE}_k = \sqrt{\frac{1}{M} \sum_{m=1}^M \|\hat{x}_{m,k} - x_k\|^2}, \quad (3.6.13)$$

where  $M$  is the number of Monte-Carlo (MC) iterations over which the root-mean is evaluated.  $\hat{x}_{m,k}$  is the filter estimate at  $k^{\text{th}}$  time-step of  $m^{\text{th}}$  MC run. Note that the truth  $x_k$  is the same throughout all MC iterations.

- (ii) **Normalized estimation error squared (NEES)**: NEES is a measure of estimator consistency which is based on the fact that a  $\chi^2$  distribution has a

mean equal to its degrees of freedom. Thus the following expression

$$\text{NEES}_k = \frac{1}{M} \left[ (\hat{\mathbf{x}}_k - \mathbf{x}_k)^T \mathbf{\Gamma}_{k|k}^{-1} (\hat{\mathbf{x}}_k - \mathbf{x}_k) \right] \quad (3.6.14)$$

should have  $n_x$  degrees of freedom (asymptotically). Since the number of MC runs are finite, the NEES for a consistent estimator should lie between the appropriate confidence bounds given by the tail probabilities of respective  $\chi^2$  density [13].

Due to long simulation time and large number of targets, both the metrics are evaluated over 200 MC runs.

### Remarks

The results in the form of RMSE and NEES plots are presented in Figs. 3.10 through Fig. 3.28. Note that target 18 does not lie within the FOV of either sensor, which is why only 19 sets of plots are present (see Fig. 3.9a). The estimated tracks for a single run using HMD-GA are shown in Fig. 3.9b. Due to high clutter, the number of tentative tracks initialized at the fusion center is clearly evident.

Based on the varying dynamics of the scenario, we have categorized the targets into three categories :

- Category A : Targets that lie within the intersection of all three sensor nodes for most of their lifetime.
- Category B : Targets that lie within the intersection of two sensor nodes.
- Category C : Targets that lie in the FOV of only one sensor. Due to memory, such tracks will be maximally correlated.

The tracks in Category A are fused by all three sensors and consist of Targets 4, 10, and 15. Due to maximum information availability, these targets have the least error relative to the error in estimation of other targets. This is also evident in the relevant plots in Figs. 3.13, 3.19 and 3.24. The NEES plots for ICI and HMD-GA show no difference, while CI, proven to be highly conservative, lies at the bottom of all NEES plots.

The distance from the node plays an important role in the error performance, as the cross-range error increases with range. Thus, as observed, Target 4 has the highest RMSE, whereas Target 10 has the least. Target 15 moves to Category B towards the end of its life, due to which the error increases. The expected error in CI, however, remains close to ICI throughout the simulation. HMD-GA has the least estimation error for most of the sample time, as evident in the plots.

The tracks in Category B consist of Targets 2, 5, 7, 8, 11, 12, 13, 14, and 16. The RMSE is moderately higher than Category A due to information availability from only two sensor nodes. Due to the upper bound on the employed track-fusion algorithms, the NEES plot also suggests a higher reported covariance than in Category A. It is obvious that recurring fusion should decrease fused covariance, due to which NEES plots are comparatively higher in the case of Category A.

The third category is interesting since these targets lie in the FOV of only one sensor. Due to memory, the fusion track contains a predicted estimate of the last track sent from the same node. If the sensor node has reached a steady state such that the track covariance is not varying enough with time, the correlation between the global track and local track will be the highest among all three categories. The correlation also strongly depends on the maneuvering index, which is a function of sample time.

Thus, with decreasing values of  $T_f$  and local sample-time, the correlation increases.

It is to be noted that in the case of unbiased and consistent local estimates, the correlation coefficient increases as the local pdf become similar to each other. The expressions for HMD and CI become, in this case,

$$\frac{p^1(\mathbf{x})p^2(\mathbf{x})}{(p^1(\mathbf{x}))^{\omega_2} (p^2(\mathbf{x}))^{\omega_1}}_{p_1 \rightarrow p_2} = \frac{p_1(\mathbf{x})p_2(\mathbf{x})}{\omega_2 p_1(\mathbf{x}) + \omega_1 p_2(\mathbf{x})}_{p_1 \rightarrow p_2} \quad (3.6.15)$$

and similarly in the case of ICI.

$$\mathbf{\Gamma}_{\text{ICI}}|_{\mathbf{\Gamma}_1 \rightarrow \mathbf{\Gamma}_2} \rightarrow \mathbf{\Gamma}_1 \quad (3.6.16)$$

Thus, the result is the local track density itself. Therefore, in the case of perfect correlation, almost no fusion takes place with conservative fusion techniques. This is shown in the NEES plot for Category C targets, where the relative conservativeness of CI in comparison to ICI and HMD-GA vanishes. All fusion techniques tend to produce a similar fused covariance, due to which the NEES plot looks similar in all three cases. Such objects are targets 3, 6, 9, 17, and 20. The corresponding RMSE plot shows the highest error among all categories, which is almost equal to the local node covariance (not shown due to space considerations).

The reported covariance in the case of HMD-GA is always less than ICI (see appendix B for proof), while being overall conservative with respect to true error. The same is reflected in the NEES result, with exceptions in the case of high cross-correlation, wherein the fused covariance is almost the same for all implemented fusers. In some cases, the NEES graph suggests higher covariance for HMD-GA, but this is not true. The result is only due to the use of finite MC samples.

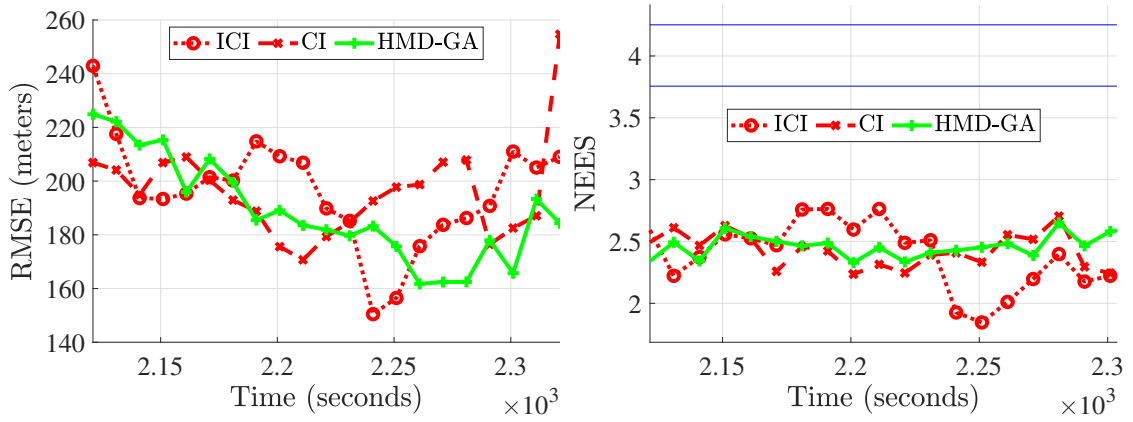


Figure 3.10: Target 1: RMSE Position and Velocity.

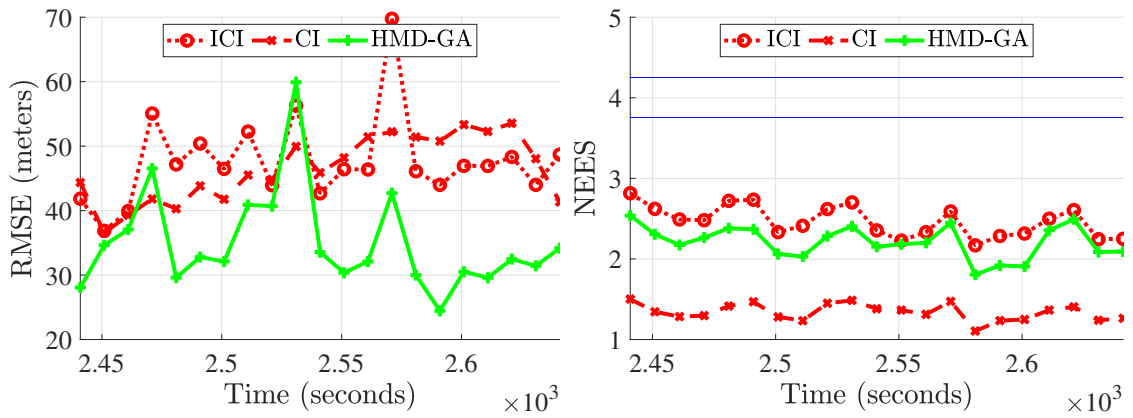


Figure 3.11: Target 2: RMSE Position and Velocity.

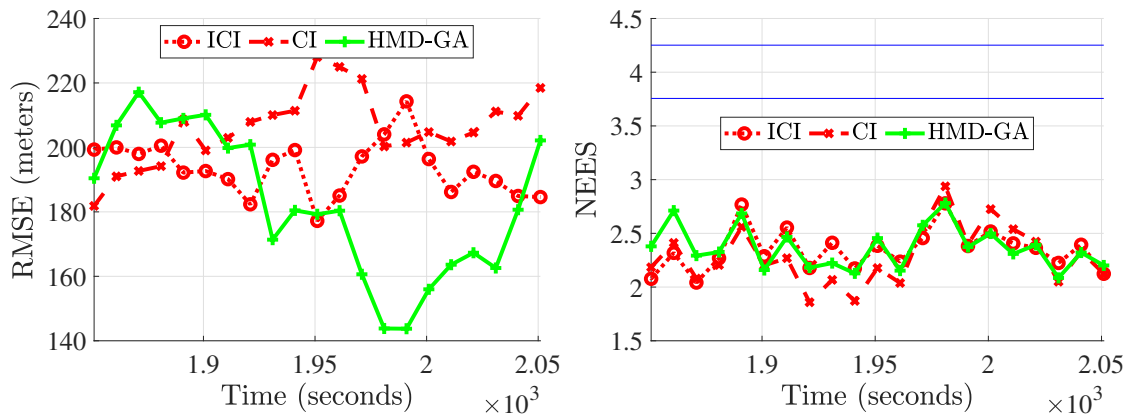


Figure 3.12: Target 3: RMSE Position and Velocity.



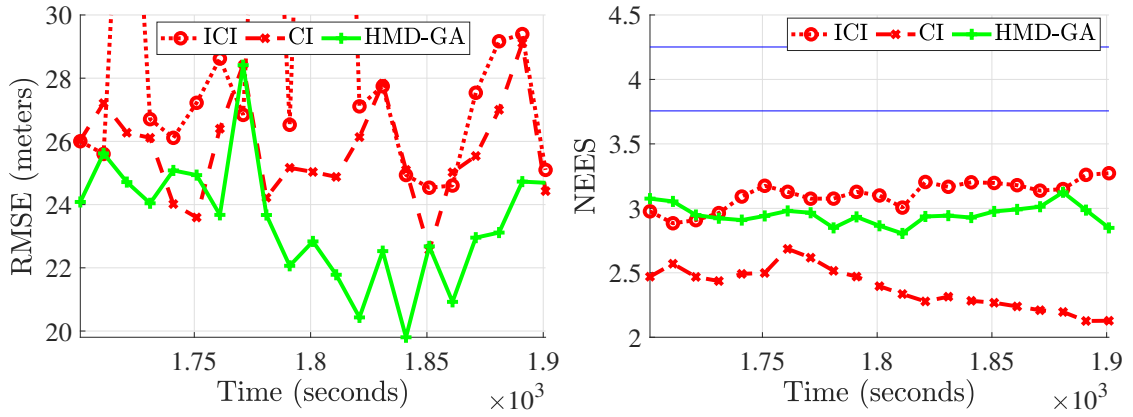


Figure 3.13: Target 4: RMSE Position and Velocity.

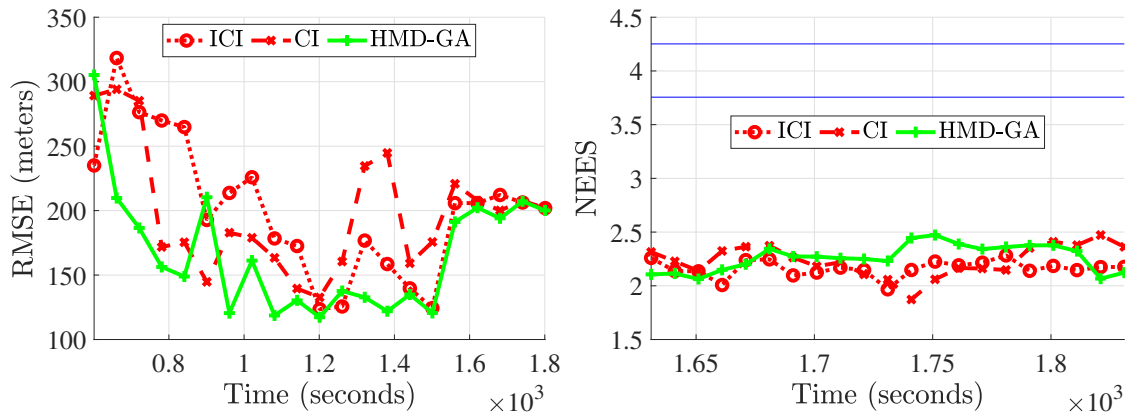


Figure 3.14: Target 5: RMSE Position and Velocity.

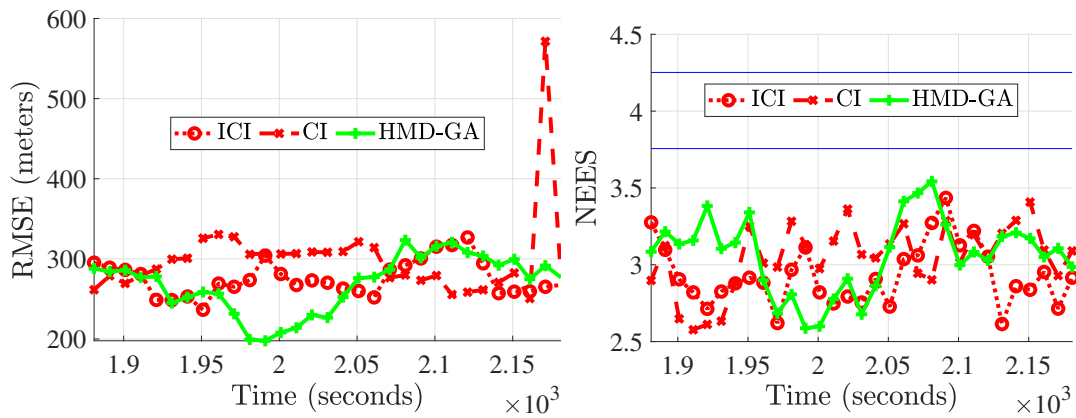


Figure 3.15: Target 6: RMSE Position and Velocity.

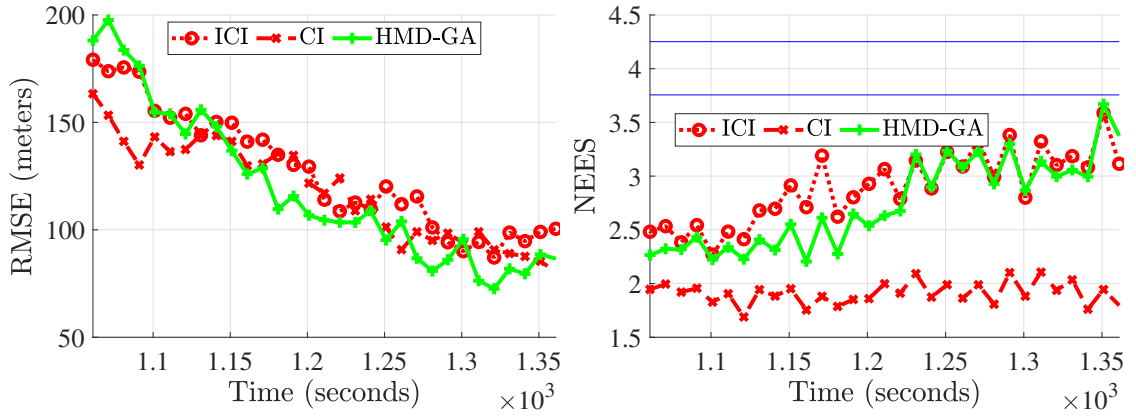


Figure 3.16: Target 7: RMSE Position and Velocity.

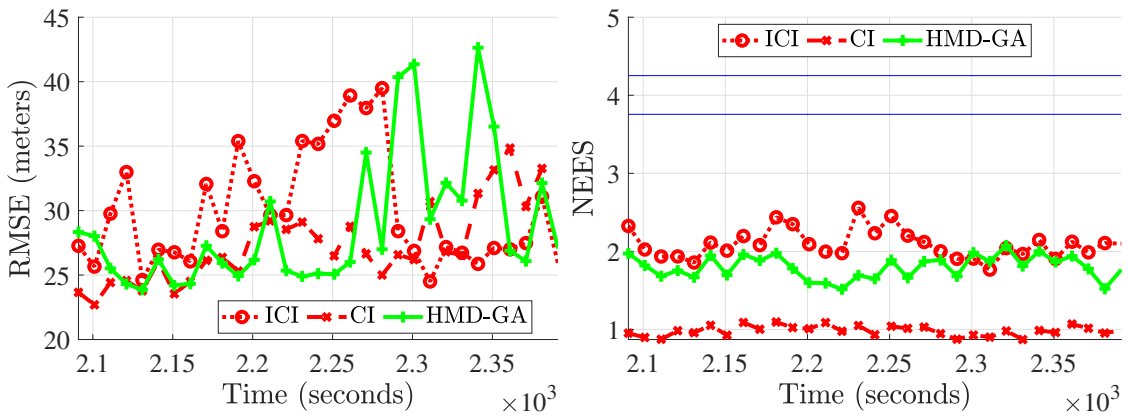


Figure 3.17: Target 8: RMSE Position and Velocity.

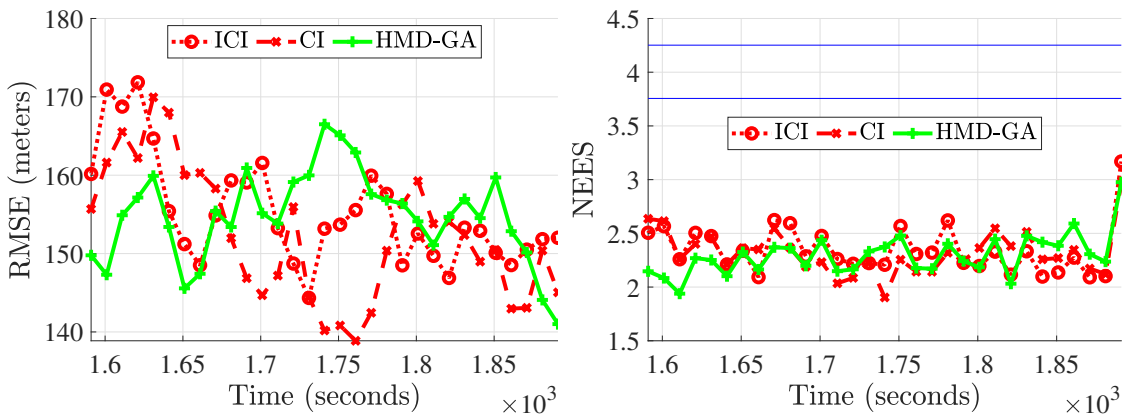


Figure 3.18: Target 9: RMSE Position and Velocity.

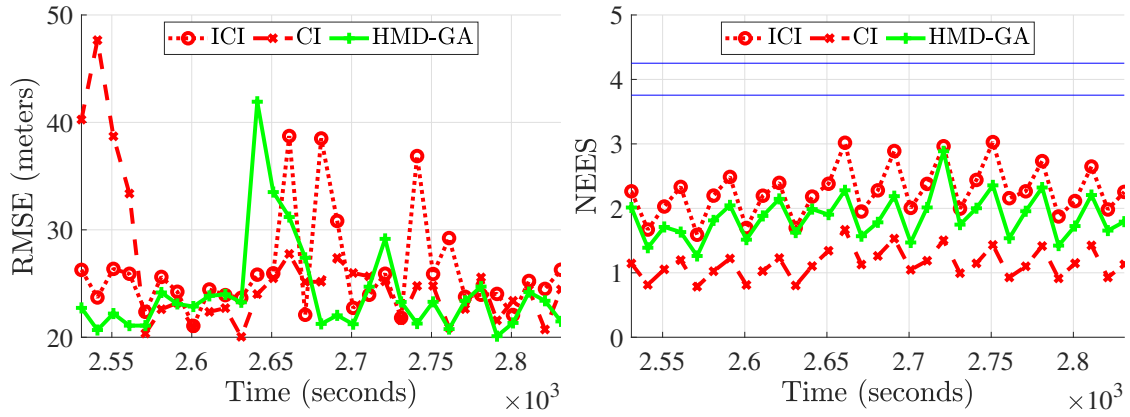


Figure 3.19: Target 10: RMSE Position and Velocity.

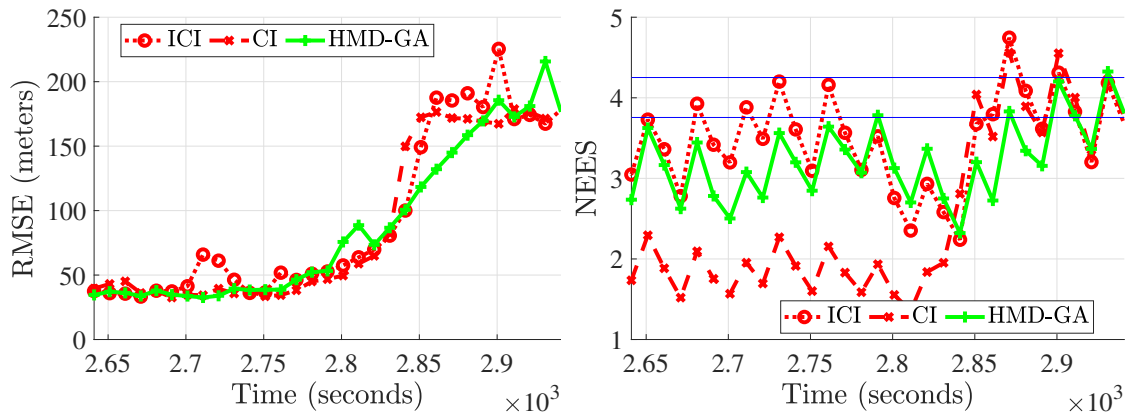


Figure 3.20: Target 11: RMSE Position and Velocity.

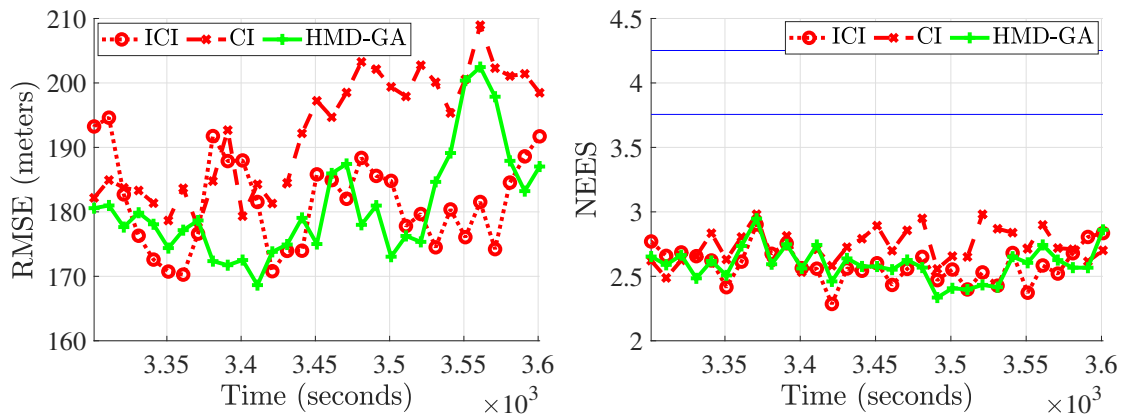


Figure 3.21: Target 12: RMSE Position and Velocity.

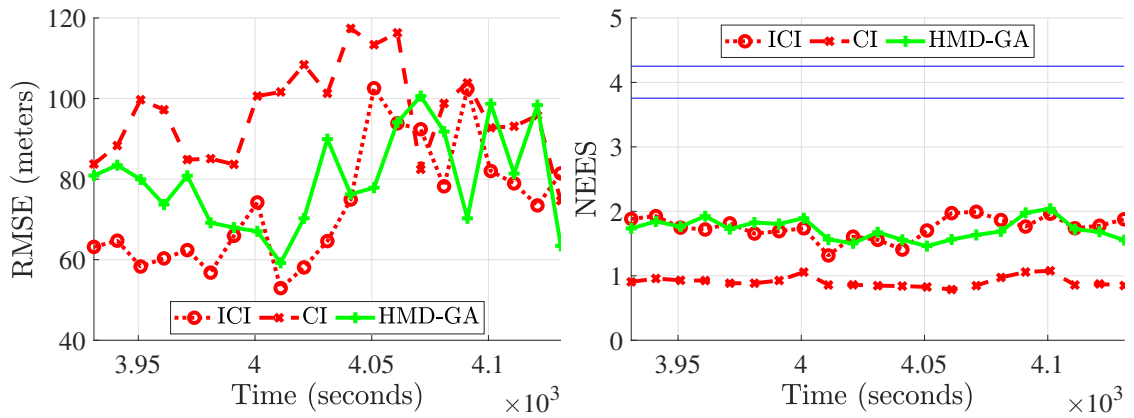


Figure 3.22: Target 13: RMSE Position and Velocity.

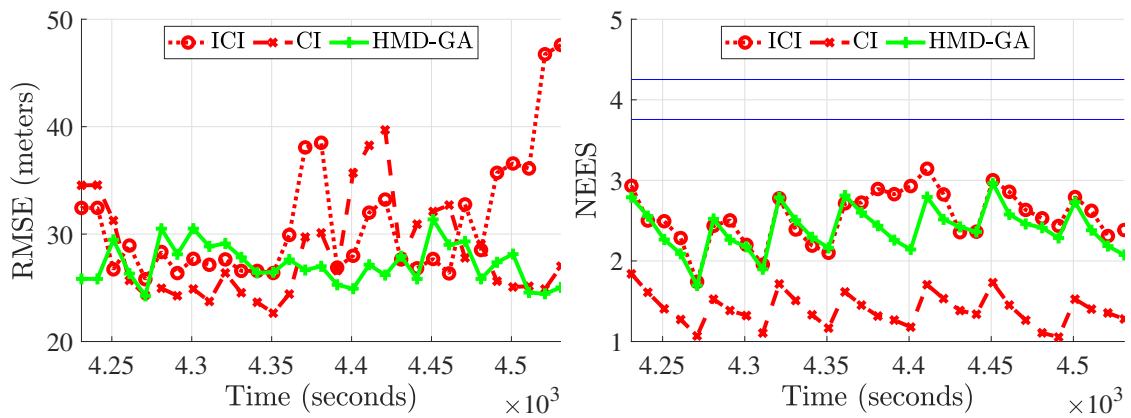


Figure 3.23: Target 14: RMSE Position and Velocity.

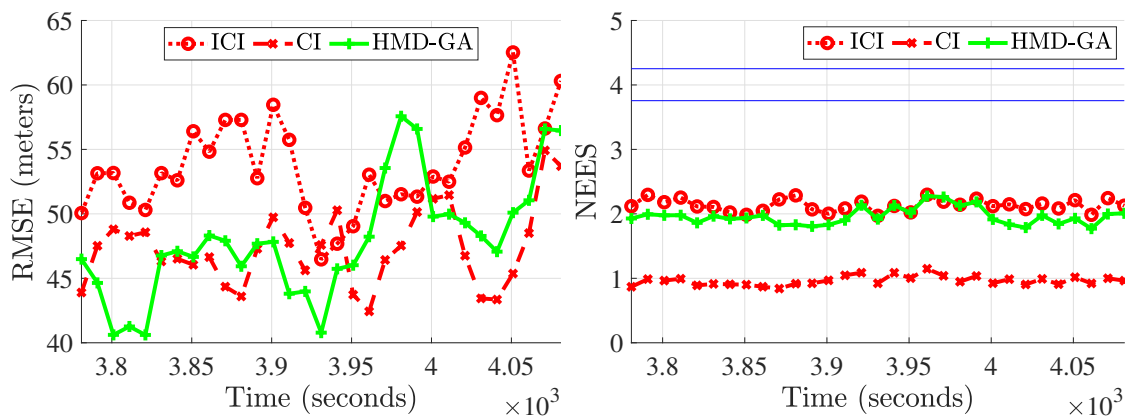


Figure 3.24: Target 15: RMSE Position and Velocity.

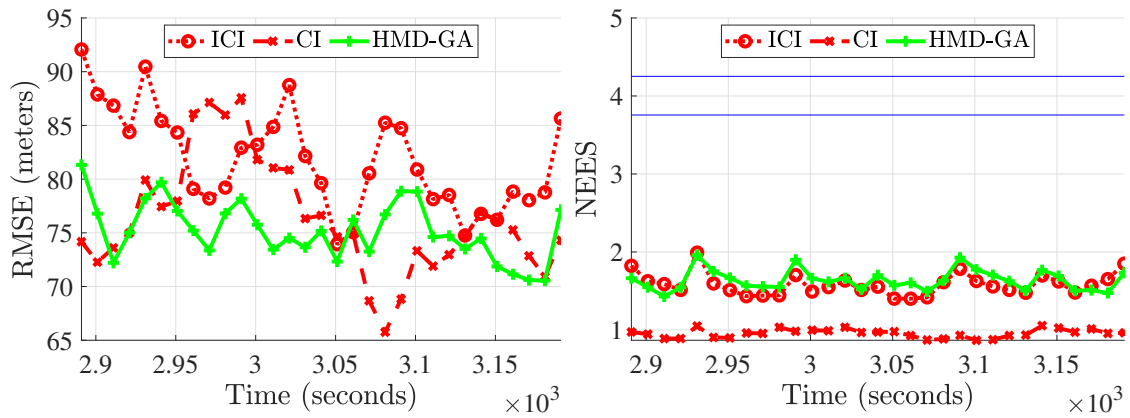


Figure 3.25: Target 16: RMSE Position and Velocity.

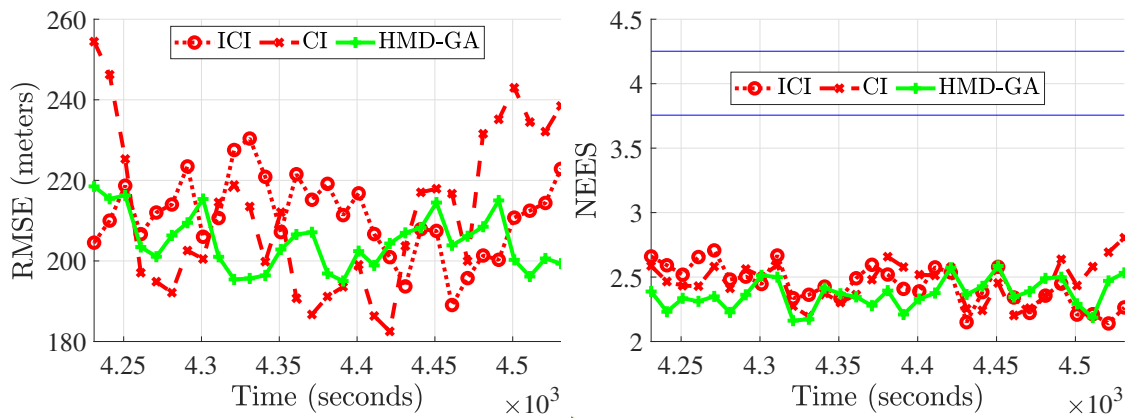


Figure 3.26: Target 17: RMSE Position and Velocity.

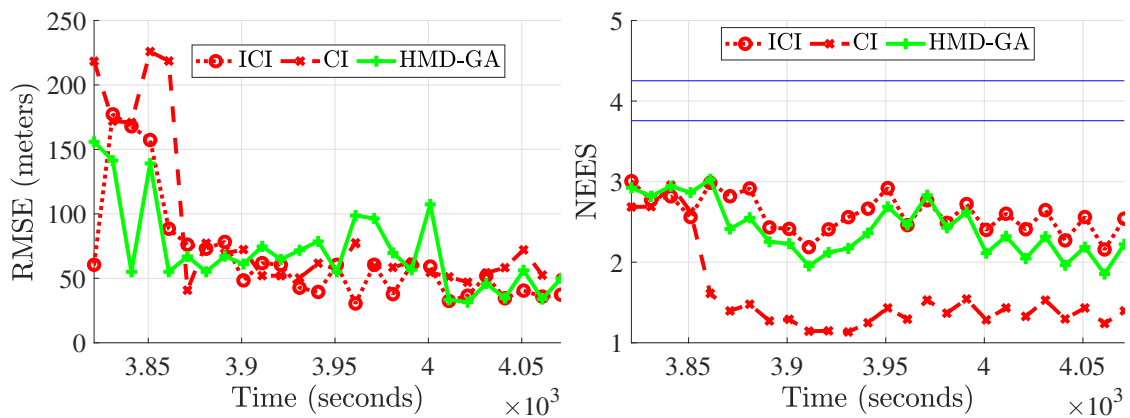


Figure 3.27: Target 19: RMSE Position and Velocity.

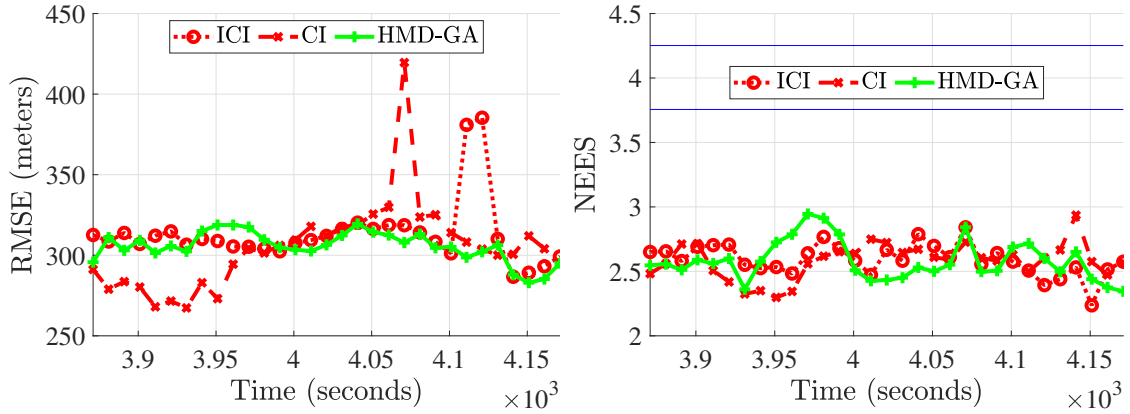


Figure 3.28: Target 20: RMSE Position and Velocity.

### 3.7 Conclusion

A deeper investigation of harmonic mean density based fusion was performed in this chapter, wherein we showed that HMD is a naturalized consensus-based fusion algorithm that minimizes a weighted  $\chi^2$  divergence between the local densities. Thus, the fusion is an agreement rather than a distributed Bayes' type result, though its expression is closely mimicked by HMD. The agreement occurs by optimizing a fused density such that it is closer to local track densities in the sense of a statistical distance ( $\chi^2$  divergence). Two versions of HMD are investigated. The first one, which was first proposed in [84] by approximating a mixture in the denominator by a Gaussian, is termed HMD-GA, and the second is HMD-S, which uses samples from either of the participating local densities, thus providing versatility in fusion.

A consistency analysis of HMD-GA has been performed and found that it depends on spread-of-means terms. In extreme cases of correlation, the HMD-GA was proven to be consistent. For intuition, consistency in the scalar case has also been proven. It is clear that the consistency of HMD-GA strongly depends on the consistency of local estimates. Due to similarity of expression, HMD-GA is closely linked with the inverse

covariance intersection, which has been closely investigated here. It was found that though ICI is considered for a specific correlation structure, it does tend to contradict the same. A Gaussian mixture implementation of ICI is also proposed in this chapter.

Sampling-based HMD implementation possesses the desirable quality of being as conservative as possible by inflating the local covariances. This might have appreciable utility in situations with anomalies and track biases. Future research will be based on such problems. Also, tests on three different simulation scenarios reveal the consistent nature of HMD-GA, with the tightest covariance among major conservative methods. The results also show that HMD-GA performs best in terms of error metrics.

The latest research on multi-sensor, multi-target tracking promotes employing extended-target mechanisms, RFS-based filtering, multi-hypothesis trackers, and anomaly-tolerant filtering on local nodes. Flexible and easy implementation of HMD-GA will definitely serve as a promising fusion methodology in all of these cases.

# Chapter 4

## Multi-Sensor Multi-Target Tracking With Missing Local Covariance

### 4.1 Abstract

In a distributed sensor fusion architecture, the local sensor nodes transmit the posterior densities characterized by mean and covariance, which are fused together to obtain the global estimate. However, in a communication constrained network, especially in warfare situations, the covariance might not be available for fusion. This chapter addresses the problem of covariance unavailability and reconstruction using parallel recursion of covariance for each local node at the fusion center. The covariance is initialized by generating converted covariance conditioned on the local posterior estimates, which is recursively propagated with time. Though there is still a requirement of a vector containing measurement update times at the local node, it



is shown that this requirement can be minimized.

The efficacy of the result is shown on a practical multi-target multi-sensor tracking scenario wherein covariance intersection is performed with reconstructed covariance. The simulations confirm that the reconstructed covariance converges with the true covariance in the sense of Frobenius norm. Results based on metrics such as root-mean-square error and normalized estimation error suggest that the existing fusion algorithms can prevail in a communication constrained network.

## 4.2 Introduction

In a distributed sensor network, each sensor platform generates its version of the target probability density conditioned upon its set of measurements. The fusion center (FC) receives the local posterior probability densities (assumed Gaussian) in the form of mean and covariance, and fuses them optimally in the Bayesian sense as [8],

$$p(\mathbf{x}_k | z_k^1 \cup z_k^2) \propto \frac{p(\mathbf{x}_k | z_k^1) p(\mathbf{x}_k | z_k^2)}{p(\mathbf{x}_k | z_k^1 \cap z_k^2)}, \quad (4.2.1)$$

where  $z_k^i$  is the measurement for sensor  $i$  and  $\mathbf{x}_k$  is the target state at time  $k$ . In parametric and symmetric distributions, the density function is fully characterized by the mean and the error covariance. Thus, transmission of these two quantities alone is necessary for fusion. As iterated in the previous chapters, distributed fusion is also termed as track-to-track fusion since tracks (state estimates) are fused together instead of measurements. This is an ideal strategy for two reasons – 1. Centralized fusion would require all the measurements (including clutter) to be transmitted to FC. 2. Distributed fusion has been proven to produce near-optimal results [24, 61, 91].

The problem of missing covariance is mainly seen in networks with constrained communication bandwidth and also in smart sensor systems [105]. Tactical information systems such as the joint tactical information distribution system (JTIDS), Link 11 and Link 16 are examples of networks with constrained bandwidth [62]. For instance, the JTIDS offers a bit rate of 115 kbps [36] which could be inadequate to transmit all elements of the covariance matrix from all the sensors in the network.

The research on the topic of covariance unavailability in distributed fusion is rather limited. In [11], the covariance was reconstructed at FC using the maneuvering index of the target and the position root-mean-square error from the local covariance matrix. It was assumed that the local nodes employ legacy trackers (steady-state filters). Upon implementation, it was found by the authors that the non-linear equation (see eqn. (16) in [11]) required to be solved for maneuvering index yields a unique solution only for very low RMS errors due to the assumption of steady state.

Another work on distributed fusion without covariance was performed by Lobbia et. al. in [62]. The covariance reconstruction was performed heuristically using a spherical error probability. The chapter attempted the problem in two dimension only. However, the generalization to higher dimension is rather cumbersome and depends on user chosen parameters. Yuan et. al. in [105] attempted to tackle covariance unavailability in automotive smart sensor systems by quoting the available local track estimate as ‘super-measurement’ and using a legacy filter for two-dimensional tracking. The yielded ‘super-estimates’ and corresponding covariance is used for fusion after decorrelation. The quality of the fuser covariance in this case is ambiguous, since it’s clear from the consistency plots (confidence levels) that very less number of Monte-Carlo iterations were used. For a communication constrained environment, a

comparison of distributed and centralized architectures is also presented in [42]. This work provides a crude upper limit on the network capacity in distributed tracking systems.

In this chapter, a solution to deal with the track fusion architecture with missing local track covariance information is proposed. The FC is assumed to have enough computational resources to perform the reconstruction of track error covariance for each sensor platform. In particular, a summary of ideas mentioned in this chapter can be enumerated as:

1. Initialization of covariance matrix at FC using state estimate, sensor covariance using converted measurement (CM) technique.
2. Generalization of the covariance initialization to non-linear sensor models using CM.
3. Development of a covariance update algorithm at FC solely based on state estimate, sensor covariance and a vector containing measurement update timestamp at local sensor.
4. Analysis of positive-definiteness for initialization of the covariance matrices.

The rest of the chapter is structured as follows; in Section 4.3, we state the problem of fusion in distributed sensors networks along with a survey of track fusion algorithms. In section 4.4, a review of the converted measurement Kalman filter for radar sensors is provided, which will be used in covariance initialization in Section 4.5. A generalization for initializing covariance matrix for any non-linear sensor model using inverse measurement Jacobian is presented in Section 4.6. The local

covariance update recursion is presented in Section 4.7. Results based on simulations are presented in Section 4.8. Finally, the article is concluded in Section 4.9.

### 4.3 Problem Formulation and Fusion Review

Let's consider a scenario with two sensor platforms –  $\mathcal{S}^1$  and  $\mathcal{S}^2$  generating processed track density  $p(\mathbf{x}_k|z_k^i)$  conditioned on current measurement  $z_k^i$  where  $k$  is the time and  $i \in \{1, 2\}$ . Assume that both tracks are conditioned on a common prior density at time  $k$  which is quantified as  $p_c(\mathbf{x}_k)$ . Then, we know

$$p(\mathbf{x}_k|z_k^i) \propto p(z_k^i|\mathbf{x}_k) \times p_c(\mathbf{x}_k) \quad (4.3.1)$$

Using naive fusion with such local density results in

$$p_N(\mathbf{x}_k|z_k^1 \cup z_k^2) \propto p(z_k^1|\mathbf{x}_k)p_c(\mathbf{x}_k)p(z_k^2|\mathbf{x}_k)p_c(\mathbf{x}_k), \quad (4.3.2)$$

where  $p_N(\mathbf{x}_k|z_k^1 \cup z_k^2)$  is the naively fused density. Thus, the common information appears twice, which leads to an optimistic estimate (e.g. multiplying two Gaussian densities results in a Gaussian density with a lower or equal covariance). A heuristic approach would be to use naive fusion while dividing once with the common information,

$$p(\mathbf{x}_k|z_k^1 \cup z_k^2) \propto \frac{p(\mathbf{x}_k|z_k^1)p(\mathbf{x}_k|z_k^2)}{p(\mathbf{x}_k|z_k^1 \cap z_k^2)} \quad (4.3.3)$$

This is (roughly) the Bayesian formulation for track fusion from two sensors. If Gaussianity is assumed, the local posterior density can be parameterized using the

corresponding mean and the covariance matrix as

$$p(\mathbf{x}_k | z_k^i) \sim \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^i, \Gamma_k^i) \quad (4.3.4)$$

Due to unavailability of the common information  $p(\mathbf{x}_k | z_k^1 \cap z_k^2)$  at the FC, a number of fusion strategies have been formulated which yield closed form expressions for fused mean and covariance under Gaussianity. Some of them are enumerated as:

1. **Naive Fusion** [14] – Naive fusion is optimal under the presumption that tracks are independent of each other. Since, this is not true due to the common prior, the *rumour* is propagated twice as shown in equation (4.3.2). The naive fused density is,

$$p_N(\mathbf{x}_k | z_k^i \cup z_k^j) \sim \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_k^N, \Gamma_k^N) \quad (4.3.5)$$

where the naive fused mean and covariance are

$$\Gamma_k^N = \left( \Gamma_k^{i-1} + \Gamma_k^{j-1} \right)^{-1} \quad (4.3.6a)$$

$$\hat{\mathbf{x}}_k^N = \Gamma_k^N \left( \Gamma_k^{i-1} \hat{\mathbf{x}}_k^i + \Gamma_k^{j-1} \hat{\mathbf{x}}_k^j \right) \quad (4.3.6b)$$

2. **Information Matrix Fusion (IMF)** [24] – IMF is a memory based fusion algorithm, which means it needs access to the previous track estimate and covariance. At full communication rate, the IMF can be proven to be equivalent to centralized track fusion [23]. Given the knowledge of previous estimates, it

also assumes a closed form under Gaussianity.

$$\Gamma_{k|k}^{IMF^{-1}} = \Gamma_{k|k-1}^{-1} + \sum_{i=1}^N \left( \Gamma_{k|k}^{i-1} - \Gamma_{k|k-1}^{i-1} \right) \quad (4.3.7a)$$

$$\begin{aligned} \Gamma_{k|k}^{IMF^{-1}} \hat{\mathbf{x}}_{k|k}^{IMF} &= \Gamma_{k|k-1}^{-1} \hat{\mathbf{x}}_{k|k-1} \\ &+ \sum_{i=1}^N \left( \Gamma_{k|k}^{i-1} \hat{\mathbf{x}}_{k|k}^i - \Gamma_{k|k-1}^{i-1} \hat{\mathbf{x}}_{k|k-1}^i \right) \end{aligned} \quad (4.3.7b)$$

where  $N$  is the number of local nodes. Extensions of IMF for non-full rate and asynchronous sensors are given in [89].

3. **Covariance Intersection** – The IMF requires the knowledge of not only the previous fused estimate but also the last local estimate. Covariance intersection attempts to resolve this problem by taking a convex combination of the estimates,

$$\Gamma_{k|k}^{CI} = \left( \omega \Gamma_{k|k}^{i-1} + (1 - \omega) \Gamma_{k|k}^{j-1} \right)^{-1} \quad (4.3.8a)$$

$$\hat{\mathbf{x}}_{k|k}^{CI} = \Gamma_{k|k}^{CI} \left( \omega \Gamma_{k|k}^{i-1} \hat{\mathbf{x}}_{k|k}^i + (1 - \omega) \Gamma_{k|k}^{j-1} \hat{\mathbf{x}}_{k|k}^j \right) \quad (4.3.8b)$$

where  $\omega \in [0, 1]$  is selected using a suitable optimization problem [98]. Covariance intersection, also known as the geometric mean density fusion [47] belongs to a family of f-mixture distributions. Using this interpretation, the algorithm can be extended (with some approximations) to Gaussian mixtures as well [43].

4. **Tracklet** [38] – Tracklets refers to the track data for a target, which is not cross-correlated with any other track in the network. For an estimate  $\hat{\mathbf{x}}_{k|k}^i$  and its covariance  $\Gamma_{k|k}^i$ , the corresponding tracklet  $\hat{\mathbf{u}}_{k|k}^i$  and its covariance  $\mathbf{U}_{k|k}^i$  are

calculated as,

$$\begin{aligned} \hat{\mathbf{u}}_{k|k}^i &= \hat{\mathbf{x}}_{k|k-1}^i \\ &+ \Gamma_{k|k-1}^i (\Gamma_{k|k-1}^i - \Gamma_{k|k}^i)^{-1} [\hat{\mathbf{x}}_{k|k}^i - \hat{\mathbf{x}}_{k|k-1}^i] \end{aligned} \quad (4.3.9a)$$

$$\mathbf{U}_{k|k}^i = \left[ \Gamma_{k|k-1}^i (\Gamma_{k|k-1}^i - \Gamma_{k|k}^i)^{-1} - \mathbf{I} \right] \Gamma_{k|k-1}^i, \quad (4.3.9b)$$

which can now be fused naively assuming independence as in equation (4.3.6).

Note that this version of tracklets assume zero process noise for the target [37].

As observed in all fusion strategies, the track covariance information is ubiquitous. This chapter attempts to use the above-mentioned techniques in situations where track covariance information is not available from local sensor platforms. Our strategy is to initialize the track covariance at the FC using converted measurement covariance which only requires local track estimates, and then use a parallel Kalman filter at the FC for each local node to update/predict the local track covariance given measurement update time-stamps from the local sensor. The local track covariance resulting from such recursions will then be used for fusion of local track densities.

In other words, we will have a parallel KF covariance recursion for each sensor node at FC. Thus, the fusion center is assumed to have sufficient computational resources. Other valid requirements in the formulations are:

- Knowledge of sensor error covariance for each node at FC.
- If the fusion does not occur at full rate, the sensor node must send a vector of measurement update time stamps since the last fusion time. An attempt will be made to minimize this requirement.

- Knowledge of target parameters for all targets, i.e., target state transition matrix and target process noise covariance.
- The sensor measurement function, for generalization to sensors other than radar.

## 4.4 Converted Measurement Kalman Filter (CMKF)

This section provides an introduction to CMKF, which is used for initializing the covariance matrix at the input. The conversion measurement method (CMM) is a convenient approach to linearize the measurements to Cartesian coordinates. CMKF is still an active area of research for over three decades since it was first published in [19].

The state transition model is expressed in Cartesian coordinates as,

$$\mathbf{x}_{k+1|k+1} = \mathbf{F}\mathbf{x}_{k+1|k} + \mathbf{w}_k \quad (4.4.1)$$

where  $\mathbf{F}$  is the state transition matrix and  $\mathbf{w}_k$  is a random vector governed by the white stationary process noise  $\sim \mathcal{N}(\mathbf{w}_k; \mathbf{0}, \mathbf{Q}_k)$ .

Considering the case of a 2D radar, the standard range azimuth model is given by

$$\mathbf{z}_k = \begin{bmatrix} r_k^m \\ \theta_k^m \end{bmatrix} = \begin{bmatrix} \sqrt{(x_k^2 + y_k^2)} \\ \tan^{-1}\left(\frac{y_k}{x_k}\right) \end{bmatrix} + \mathbf{v}_k \quad (4.4.2)$$

where  $\mathbf{v}_k$  is the sensor noise governed by  $\sim \mathcal{N}(\mathbf{v}_k; \mathbf{0}, \mathbf{R}_k)$ , uncorrelated with  $\mathbf{w}_k$ . The



nominal polar to Cartesian conversion suggests the following transformation,

$$x_k^m = r_k^m \cos(\theta_k^m), \quad y_k^m = r_k^m \sin(\theta_k^m) \quad (4.4.3)$$

However, the unbiased condition states,

$$\mathbb{E}[x_k] = \mathbb{E}[x_k^m] = \mathbb{E}[r_k^m \cos(\theta_k^m)] \quad (4.4.4)$$

$x_k^m$  and  $y_k^m$  are the converted coordinates using measurements  $r_k^m$  &  $\theta_k^m$  and  $x_k$  is the true state (position). Assuming independence of measurements of  $r^m$  and  $\theta^m$ , it can be shown that [14],

$$\mathbb{E}[r_k^m \cos(\theta_k^m)] = x_k e^{-\sigma_\theta^2/2} \quad (4.4.5a)$$

$$\mathbb{E}[r_k^m \sin(\theta_k^m)] = y_k e^{-\sigma_\theta^2/2} \quad (4.4.5b)$$

Therefore the unbiased transformation is (using  $\lambda$  for the bias factor)

$$x_k^m = \lambda^{-1} r_k^m \cos(\theta_k^m) \quad (4.4.6a)$$

$$y_k^m = \lambda^{-1} r_k^m \sin(\theta_k^m) \quad (4.4.6b)$$

The corresponding covariance should be (ideally),

$$\mathbf{R}_k^c = \begin{bmatrix} \text{var}(x_k) & \text{cov}(x_k, y_k) \\ \text{cov}(y_k, x_k) & \text{var}(y_k) \end{bmatrix} \quad (4.4.7)$$

This would require true target state to be available, hence the covariance is obtained

by conditioning on measurements,

$$\mathbf{R}^m = \begin{bmatrix} \text{var}(x_k | r_k^m, \theta_k^m) & \text{cov}(x_k, y_k | r_k^m, \theta_k^m) \\ \text{cov}(y_k, x_k | r_k^m, \theta_k^m) & \text{var}(y_k | r_k^m, \theta_k^m) \end{bmatrix} \quad (4.4.8)$$

The elements of the converted covariance matrix  $\mathbf{R}_k^m$  conditioned on the unbiased measurements are [14] (dropping time-index  $k$  for clarity),

$$\begin{aligned} R^m(1, 1) &= [\lambda^{-2} - 2](r^m)^2 \cos^2(\theta^m) + 0.5\{(r^m)^2 + \sigma_r^2\} \\ &\quad \times \{1 + \lambda^4 \cos(2\theta^m)\} \end{aligned} \quad (4.4.9a)$$

$$\begin{aligned} R^m(2, 2) &= [\lambda^{-2} - 2](r^m)^2 \sin^2(\theta^m) + 0.5\{(r^m)^2 + \sigma_r^2\} \\ &\quad \times \{1 - \lambda^4 \cos(2\theta^m)\} \end{aligned} \quad (4.4.9b)$$

$$\begin{aligned} R^m(1, 2) &= 0.5 [\lambda^{-2}(r^m)^2 + \{(r^m)^2 + \sigma_r^2\}\lambda^4 - 2(r^m)^2] \\ &\quad \times \sin(2\theta^m) \end{aligned} \quad (4.4.9c)$$

The new *linearized* radar model is,

$$\mathbf{z}_k = \begin{bmatrix} \lambda^{-1} r_k^m \cos(\theta_k^m) \\ \lambda^{-1} r_k^m \sin(\theta_k^m) \end{bmatrix} + \mathbf{v}'_k \quad (4.4.10)$$

where,  $\mathbf{v}'_k \sim \mathcal{N}(\mathbf{v}_k; \mathbf{0}, \mathbf{R}_k^m)$  is now the measurement noise in Cartesian coordinates. A standard linear Kalman filter can now be used with the converted measurements.

For three dimensions radar tracking, the framework of unbiased measurements remain the same with additional calculations for elevation. Readers are referred to [63] for accurate debiasing of measurements and [59] for a comparison with EKF and analysis.

## 4.5 Covariance Initialization at the Fusion Center

When the fusion center receives the state estimate for a local node, the first step in covariance construction is to initialize it. In this section, we propose the initialization using the state estimate and the knowledge of sensor parameters. Some approximation needs to be introduced, as we do not have complete information about the posterior density. The following derivations are provided for two-dimensional range-azimuth sensors, but can easily be extended. In later sections, we will provide a generalized approach using Jacobians.

At FC, suppose we have the posterior estimate  $\hat{\mathbf{x}}_{k|k}^i$  from sensor node  $i$  at time  $k$ . Then, two-point initialization using the CMM discussed in section 4.4 could be used to initialize the covariance matrix. Since the local measurements are not available, it is imperative to use the converted covariance formulation conditioned on the available information. One possible way is,

$$\mathbf{R}_k^t = \begin{bmatrix} \text{var}(x|x_{k|k}^i, y_{k|k}^i) & \text{cov}(x, y|x_{k|k}^i, y_{k|k}^i) \\ \text{cov}(y, x|x_{k|k}^i, y_{k|k}^i) & \text{var}(y|x_{k|k}^i, y_{k|k}^i) \end{bmatrix} \quad (4.5.1)$$

The exact covariance formulation of  $\mathbf{R}_k^t$  for two-dimensional sensors with error standard deviation in range and azimuth as  $\sigma_{r,i}$  and  $\sigma_{\theta,i}$  respectively, as given in [22] is

(dropping time-index),

$$\begin{aligned} \mathbf{R}_k^t(1, 1) &= \frac{1}{2} \left( r_t^{i2} + \sigma_{r^i}^2 + \sigma_{r_t^i}^2 \right) \left[ 1 + \cos(2\theta_t^i) e^{-2\sigma_{\theta^i}^2} e^{-2\sigma_{\theta_t^i}^2} \right] e^{\sigma_{\theta^i}^2} \\ &\quad - \frac{1}{2} \left( r_t^{i2} + \sigma_{r_t^i}^2 \right) \left[ 1 + \cos(2\theta_t^i) e^{-2\sigma_{\theta_t^i}^2} \right] \end{aligned} \quad (4.5.2a)$$

$$\begin{aligned} \mathbf{R}_k^t(2, 2) &= \frac{1}{2} \left( r_t^{i2} + \sigma_{r^i}^2 + \sigma_{r_t^i}^2 \right) \left[ 1 - \cos(2\theta_t^i) e^{-2\sigma_{\theta^i}^2} e^{-2\sigma_{\theta_t^i}^2} \right] e^{\sigma_{\theta^i}^2} \\ &\quad - \frac{1}{2} \left( r_t^{i2} + \sigma_{r_t^i}^2 \right) \left[ 1 - \cos(2\theta_t^i) e^{-2\sigma_{\theta_t^i}^2} \right] \end{aligned} \quad (4.5.2b)$$

$$\begin{aligned} \mathbf{R}_k^t(1, 2) &= \frac{1}{2} \left( r_t^{i2} + \sigma_{r^i}^2 + \sigma_{r_t^i}^2 \right) \left[ \sin(2\theta_t^i) e^{-2\sigma_{\theta^i}^2} e^{-2\sigma_{\theta_t^i}^2} \right] e^{\sigma_{\theta^i}^2} \\ &\quad - \frac{1}{2} \left( r_t^{i2} + \sigma_{r_t^i}^2 \right) \left[ \sin(2\theta_t^i) e^{-2\sigma_{\theta_t^i}^2} \right] \end{aligned} \quad (4.5.2c)$$

The range ( $r_t$ ) and azimuth ( $\theta_t$ ) estimate are obtained from the posterior estimate from the sensor node.

$$r_{t,k}^i = \sqrt{(\hat{x}_{k|k}^i)^2 + (\hat{y}_{k|k}^i)^2} \quad (4.5.3a)$$

$$\theta_{t,k}^i = \tan^{-1} \left( \frac{\hat{y}_{k|k}^i}{\hat{x}_{k|k}^i} \right) \quad (4.5.3b)$$

$\sigma_{r_{t,k}^i}^2$  and  $\sigma_{\theta_{t,k}^i}^2$  are respectively the estimated range and azimuth error variance given by (removing time-index for clarity),

$$\sigma_{r_{t,k}^i}^2 = \frac{\text{var}(\hat{x}^i)(\hat{x}^i)^2 + \text{var}(\hat{y}^i)(\hat{y}^i)^2 + 2\text{cov}(\hat{x}^i\hat{y}^i)\hat{x}^i\hat{y}^i}{(\hat{x}^i)^2 + (\hat{y}^i)^2} \quad (4.5.4a)$$

$$\sigma_{\theta_{t,k}^i}^2 = \frac{\text{var}(\hat{x}^i)(\hat{y}^i)^2 + \text{var}(\hat{y}^i)(\hat{x}^i)^2 - 2\text{cov}(\hat{x}^i\hat{y}^i)\hat{x}^i\hat{y}^i}{((\hat{x}^i)^2 + (\hat{y}^i)^2)^2} \quad (4.5.4b)$$

The quantities  $\text{var}(\cdot)$  and  $\text{cov}(\cdot)$  in equation (4.5.4) are the elements of the covariance of the posterior estimate and hence not available. However, the available options are

- The local node transmits  $\sigma_{r_t}^2$  and  $\sigma_{\theta_t}^2$ . Or,
- Calculate the converted measurement covariance matrix by ignoring the terms containing  $\sigma_{r_t}^2$  and  $\sigma_{\theta_t}^2$ , and scale the resulting matrix appropriately using a fudge factor  $\eta$ .

$$\mathbf{R}_k^t \approx \eta \bar{\mathbf{R}}_k^t \quad (4.5.5)$$

Where,  $\bar{\mathbf{R}}_k^t$  is obtained by setting  $\sigma_{r_t}^2$  and  $\sigma_{\theta_t}^2$  to zero. For two-dimensional range-azimuth sensors, the approximate value of  $\eta$  was found to be (see Appendix C),

$$\eta = \left[ 1 - \frac{\bar{R}^t(1,1) - \bar{R}^t(2,2)}{\bar{R}^t(1,1) + \bar{R}^t(2,2)} \frac{(e^{\sigma_{\theta^i}^2} - 1)}{\cos(2\theta_t^i)(e^{-\sigma_{\theta^i}^2} - 1)} \right] \times \left[ \frac{r_t^2(e^{\sigma_{\theta^i}^2} - 1)}{2\{\bar{R}^t(1,1) + \bar{R}^t(2,2)\}} \right] + 1 \quad (4.5.6)$$

Note that, again, the subscript  $k$  has been dropped for brevity. To verify the consistency of the resulting covariance matrix obtained, a simulation was set up to evaluate normalized estimation-error squared (NEES) [21]. The posterior state estimate was sampled from a Gaussian distribution with mean as ground truth and covariance  $\Gamma$ . Such that,

$$\Gamma = \begin{bmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix} \quad (4.5.7)$$

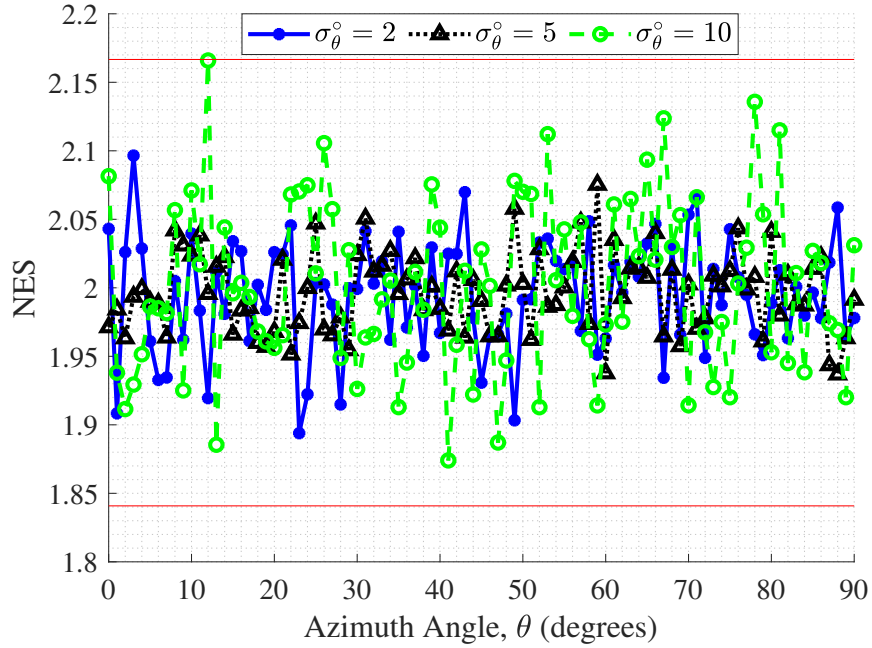


Figure 4.1: Normalized Error Squared (NES) based on  $10^3$  Monte-Carlo iterations.

where  $\sigma_x = \sigma_y = 50\text{m}$  and the correlation  $\rho$  was kept at 0.1. The range was fixed at 10,000 m, but the azimuth was varied from  $0^\circ$  to  $90^\circ$ . The results are plotted in Fig. 4.1. Standard deviations  $\sigma_r$  is kept at 50m, and  $\sigma_\theta$  at  $2^\circ$ ,  $5^\circ$  and  $10^\circ$ . The figure also shows two-sided 99% confidence bounds for  $10^3$  Monte-Carlo iterations. Moreover, it was found in this case that for low values of  $\sigma_\theta$  ( $\leq 1^\circ$ ), the matrix was consistent for  $n \approx 1$ . It is evident that the overall converted covariance matrix is fairly consistent for the mentioned values of  $\sigma_\theta$ . However, the individual components were not consistent.

Further, when the converted measurement covariance is obtained, then for a single coordinate position-velocity state  $[x, \dot{x}]$ , the covariance matrix generated using two

point initialization with  $T$  as the sampling interval looks like [14],

$$\text{cov}_k(x, \dot{x}) = \begin{bmatrix} R_k^t(1, 1), & R_k^t(1, 1)/T \\ R_k^t(1, 1)/T & 2R_k^t(1, 1)/T^2 \end{bmatrix} \quad (4.5.8)$$

and similarly for  $[y, \dot{y}]$  (and  $[z, \dot{z}]$  in case of 3-D tracks). Note that the  $2R_k^t(1, 1)/T^2$  term asserts that the covariance  $R_k^t(1, 1)$  remains unchanged since the previous time step. This, however, is not the case but can be assumed if  $r_t^i$  and  $\theta_t^i$  are varying slowly.

For three-dimensional state  $[x, \dot{x}, y, \dot{y}, z, \dot{z}]$  employing nearly constant velocity (NCV) motion model, the initialized covariance becomes the following block matrix,

$$\mathbf{P}_{k|k} = \text{diag} [\text{cov}_k(x, \dot{x}), \text{cov}_k(y, \dot{y}), \text{cov}_k(z, \dot{z})] \quad (4.5.9)$$

In the case of nearly constant acceleration (NCA), Wiener process models, we can proceed with two options:

- Assuming slow varying  $r_t$  and  $\theta_t$  and temporally uncorrelated track estimation errors, one can assume that  $\hat{\mathbf{x}}_{k|k}$  is a result of three point initialization. Then it can be proved that,

$$\text{var}(\ddot{x}) = \frac{2R_k^t(1, 1)}{T^4} \quad (4.5.10)$$

and similarly for  $\text{var}(\ddot{y})$  and  $\text{var}(\ddot{z})$ . The other option is,

- Assume an upper bound on maximum acceleration  $\ddot{x}_{\max}$ , and use  $\text{var}(\ddot{x}) = 0.25\ddot{x}_{\max}^2$ . This approach, however simple, can lead to inconsistent track covariance initialization [14].

Thus, the track covariance for NCV and NCA models can be initialized using the knowledge of posterior track estimates and sensor parameters. Closely observing the converted measurement covariance  $\mathbf{R}_k^t$ , it can be seen that the covariance between  $x$  and  $y$  is non-zero (for  $\theta_t \neq \{0^\circ, 180^\circ\}$ ). On the other hand, the covariance between a coordinate and its velocity/acceleration component is also non-zero. Thus, it is possible to initialize a full covariance matrix for such models by incorporating the cross-covariance terms. The intricacies of using this approach are discussed next.

#### 4.5.1 Full Covariance Matrix Initialization

Continuing the track covariance formulation discussed in the previous section, there is a possibility for initializing a non-sparse covariance matrix for the aforementioned models. Consider the following two-coordinate (NCV model) initial covariance matrix

$$\mathbf{P}_{k|k} = \begin{bmatrix} R_k^t(1,1) & R_k^t(1,1)/T & 0 & 0 \\ R_k^t(1,1)/T & 2R_k^t(1,1)/T^2 & 0 & 0 \\ 0 & 0 & R_k^t(2,2) & R_k^t(2,2)/T \\ 0 & 0 & R_k^t(2,2)/T & 2R_k^t(2,2)/T^2 \end{bmatrix} \quad (4.5.11)$$

The positional cross-covariance  $\text{cov}(x, y)$  is available from  $\mathbf{R}_k^t$  in the form of  $\mathbf{R}_k^t(1, 2)$  which can be incorporated. This also leads to the following developments:

- $x$  is correlated with  $y$  as per  $R_k^t(1, 2)$ . According to the initialized covariance matrix,  $x$  is also correlated with  $\dot{x}$  as  $R_k^t(1, 1)/T$  and similarly,  $y$  with  $\dot{y}$ . This means  $\dot{x}$  must also be correlated with  $\dot{y}$ .
- Using this relation,  $x$  must also be correlated with  $\dot{y}$  and vice versa.
- Similar correlation for acceleration with position and velocity can be proved.



These correlations can easily be derived under valid assumptions.

### Correlation between $\dot{x}$ and $\dot{y}$

We attempt to calculate,

$$\text{cov}(\dot{x}, \dot{y})_k = \mathbb{E} \left[ (\dot{x}_k - \hat{\dot{x}}_k)(\dot{y}_k - \hat{\dot{y}}_k)^T \right] \quad (4.5.12)$$

Using linear interpolation in the two point initialization, equation (4.5.12) can be written in terms of position coordinates,

$$\dot{x} = \frac{x_k - x_{k-1}}{T} \quad (4.5.13)$$

and same for  $y$ . This implies,

$$\text{cov}(\dot{x}, \dot{y}) = \frac{1}{T^2} \mathbb{E} \left[ (\tilde{x}_k - \tilde{x}_{k-1})(\tilde{y}_k - \tilde{y}_{k-1})^T \right] \quad (4.5.14)$$

where  $\tilde{x} = x - \hat{x}$ . It follows that

$$\text{cov}(\dot{x}, \dot{y}) = \frac{1}{T^2} \{R_{12}(k) + R_{12}(k-1)\} \quad (4.5.15)$$

$$\approx \frac{1}{T^2} \{2R_{12}(k)\} \quad (4.5.16)$$

where the following two assumption has been used

$$\mathbb{E}[\tilde{x}_k \tilde{y}_j] = \delta_{kj} R_{12}(k), \quad (4.5.17)$$

and

$$R_{12}(k) \approx R_{12}(k-1) \quad (4.5.18)$$

Table 4.1: Elements of an initialized track covariance matrix.  $R_{ij}$  refers to corresponding element in sensor covariance  $\mathbf{R}$ .

	$x$	$\dot{x}$	$\ddot{x}$	$y$	$\dot{y}$	$\ddot{y}$	$z$	$\dot{z}$	$\ddot{z}$
$x$	$R_{11}^t$	$R_{11}^t/T$	$R_{11}^t/T^2$	$R_{12}^t$	$R_{12}^t/T$	$R_{12}^t/T^2$	$R_{13}^t$	$R_{13}^t/T$	$R_{13}^t/T^2$
$\dot{x}$	$R_{11}^t/T$	$2R_{11}^t/T^2$	$2R_{11}^t/T^3$	$R_{12}^t/T$	$2R_{12}^t/T^2$	$2R_{12}^t/T^3$	$R_{13}^t/T$	$2R_{13}^t/T^2$	$2R_{13}^t/T^3$
$\ddot{x}$	$R_{11}^t/T^2$	$2R_{11}^t/T^3$	$4R_{11}^t/T^4$	$R_{12}^t/T^2$	$2R_{12}^t/T^3$	$4R_{12}^t/T^4$	$R_{13}^t/T^2$	$2R_{13}^t/T^3$	$4R_{13}^t/T^4$
$y$	$R_{21}^t$	$R_{21}^t/T$	$R_{21}^t/T^2$	$R_{22}^t$	$R_{22}^t/T$	$R_{22}^t/T^2$	$R_{23}^t$	$R_{23}^t/T$	$R_{23}^t/T^2$
$\dot{y}$	$R_{21}^t/T$	$2R_{21}^t/T^2$	$2R_{21}^t/T^3$	$R_{22}^t/T$	$2R_{22}^t/T^2$	$2R_{22}^t/T^3$	$R_{23}^t/T$	$2R_{23}^t/T^2$	$2R_{23}^t/T^3$
$\ddot{y}$	$R_{21}^t/T^2$	$2R_{21}^t/T^3$	$4R_{21}^t/T^4$	$R_{22}^t/T^2$	$2R_{22}^t/T^3$	$4R_{22}^t/T^4$	$R_{23}^t/T^2$	$2R_{23}^t/T^3$	$4R_{23}^t/T^4$
$z$	$R_{31}^t$	$R_{31}^t/T$	$R_{31}^t/T^2$	$R_{32}^t$	$R_{32}^t/T$	$R_{32}^t/T^2$	$R_{33}^t$	$R_{33}^t/T$	$R_{33}^t/T^2$
$\dot{z}$	$R_{31}^t/T$	$2R_{31}^t/T^2$	$2R_{31}^t/T^3$	$R_{32}^t/T$	$2R_{32}^t/T^2$	$2R_{32}^t/T^3$	$R_{33}^t/T$	$2R_{33}^t/T^2$	$2R_{33}^t/T^3$
$\ddot{z}$	$R_{31}^t/T^2$	$2R_{31}^t/T^3$	$4R_{31}^t/T^4$	$R_{32}^t/T^2$	$2R_{32}^t/T^3$	$4R_{32}^t/T^4$	$R_{33}^t/T^2$	$2R_{33}^t/T^3$	$4R_{33}^t/T^4$

### Correlation between $x$ and $\dot{y}$

Under similar assumptions,

$$\text{cov}(x, \dot{y})_k = \mathbb{E} \left[ (x_k - \hat{x}_k)(\dot{y}_k - \hat{\dot{y}}_k)^T \right] \quad (4.5.19)$$

$$= \frac{1}{T} R_{12}(k) \quad (4.5.20)$$

$$= \text{cov}(y, \dot{x})_k$$

## Other correlations

Table 4.1 generalizes the correlations occurring in the initialization of track covariance matrices in NCA and NCV motion models.

### 4.5.2 Positive-Definiteness

Since the cross-correlation (non-diagonal) terms can be negative in a covariance matrix, the initialization of track covariance may turn out to be non-positive definite if some correlations in the initialized covariance matrix are ignored. Consider the following example of a two-dimensional converted sensor covariance to support this statement,

$$\mathbf{R}_k^t = \begin{bmatrix} 2296.966 & -664.19 \\ -664.19 & 302.43 \end{bmatrix}$$

The initial covariance matrix according to the formulation in previous subsection for  $T = 1$  sec is (with some elements/correlations treated as 0)

$$\mathbf{P}_{k|k} = \begin{bmatrix} 2296.966 & 2296.966 & -664.19 & 0 \\ 2296.966 & 4593.2 & 0 & 0 \\ -664.19 & 0 & 302.4 & 302.4 \\ 0 & 0 & 302.4 & 604.9 \end{bmatrix}$$

whose minimum eigen value is  $-149.5$  and hence clearly non-positive definite. But the following matrix is positive definite

$$\mathbf{P}_{k|k} = \begin{bmatrix} 2296.966 & 2296.966 & -664.19 & -664.19 \\ 2296.966 & 4593.2 & -664.19 & -1328.4 \\ -664.19 & -664.19 & 302.4 & 302.4 \\ -664.19 & -1328.4 & 302.4 & 604.9 \end{bmatrix}$$

with the minimum eigen value of  $38.8$ . It can be proved that such a structure with full covariance (derived from  $\mathbf{R}_k^t$ ) always result in a positive semi-definite matrix.

**Proposition 4.5.1.** *Consider the following positive semi-definite sensor covariance matrix*

$$\mathbf{R} = \begin{bmatrix} a & b \\ b & c \end{bmatrix} \quad (4.5.21)$$

then the following matrix  $\mathbf{P}$ , derived from the elements of  $\mathbf{R}$  is always positive semi-definite.

$$\mathbf{P} = \begin{bmatrix} a & a/T & b & b/T \\ a/T & 2a/T^2 & b/T & 2b/T^2 \\ b & b/T & c & c/T \\ b/T & 2b/T^2 & c/T & 2c/T^2 \end{bmatrix} \quad (4.5.22)$$

*Proof.* First we observe that, the matrix  $\mathbf{P}$  can be written as the Kronecker product of the following matrices:

$$\mathbf{P} = \begin{bmatrix} a & b \\ b & c \end{bmatrix} \otimes \begin{bmatrix} 1 & 1/T \\ 1/T & 2/T^2 \end{bmatrix}. \quad (4.5.23)$$

Using the property for Kronecker products, we know that [107],

$$\{\text{eig}(\mathbf{M} \otimes \mathbf{N})\} = \{\text{eig}(\mathbf{M})\text{eig}(\mathbf{N})^T\}, \quad (4.5.24)$$

where  $\text{eig}(\mathbf{M})$  is a vector of eigen values of  $\mathbf{M}$  and  $\{.\}$  represents a set. Thus, the

matrix  $\mathbf{P}$  is PSD if both matrices on the R.H.S of equation (4.5.23) are positive-definite. Since from the structure it is clear that,

$$\det \left( \begin{bmatrix} 1 & 1/T \\ 1/T & 2/T^2 \end{bmatrix} \right) = 1/T^2 > 0 \quad \forall T \in \mathbb{R}, \quad (4.5.25)$$

where  $\det(\cdot)$  stands for determinant and  $\mathbf{R}$  is positive semi-definite, It is proved that  $\mathbf{P}$  has all non-negative eigen values. This proof can also be extended to 2D-NCA models where  $P$  now becomes the block matrix,

$$\mathbf{P} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix} \quad (4.5.26)$$

where,

$$\mathbf{A} = \begin{bmatrix} a & a/T & a/T^2 \\ a/T & 2a/T^2 & 2a/T^3 \\ a/T^2 & 2a/T^3 & 4a/T^4 \end{bmatrix}; \mathbf{B} = \begin{bmatrix} b & b/T & b/T^2 \\ b/T & 2b/T^2 & 2b/T^3 \\ b/T^2 & 2b/T^3 & 4b/T^4 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} c & c/T & c/T^2 \\ c/T & 2c/T^2 & 2c/T^3 \\ c/T^2 & 2c/T^3 & 4c/T^4 \end{bmatrix}.$$

Re-writing  $\mathbf{P}$  in terms of Kronecker product as above,

$$\mathbf{P} = \begin{bmatrix} a & b \\ b & c \end{bmatrix} \otimes \begin{bmatrix} 1 & 1/T & 1/T^2 \\ 1/T & 2/T^2 & 2/T^3 \\ 1/T^2 & 2/T^3 & 4/T^4 \end{bmatrix}. \quad (4.5.27)$$

From which, the determinant of second matrix on R.H.S is  $2/T^6$  which is positive for all  $T$ . Hence in this case as well,  $\mathbf{P}$  is at least positive semi-definite. In a similar manner, initial track covariance matrix for 3-D NCA and 3-D NCV model can be proved to be PSD. However, the proof is not presented in this work for brevity.  $\square$

## 4.6 Generalizing Covariance Initialization

The only caveat of using CMM is that its mechanism is designed for only range-elevation-azimuth-range-rate sensor models, which is only applicable to target tracking and localization applications. However, it is possible to extend the initialization framework discussed in Section 4.5 to a general non-linear sensor *if* its model and noise parameters are known to FC.

We use the Jacobian of the inverse sensor function to linearize the measurement model with respect to the state transition process. The technique is not entirely novel and can be developed as a version of the Extended Kalman Filter (EKF) [59].

### 4.6.1 Generalizing CMM using First Order Linearization.

Let the local sensor model be defined by the following equation,

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \quad (4.6.1)$$

where  $\mathbf{x}_k \in \mathbb{R}^{n_x}$  and  $\mathbf{z}_k \in \mathbb{R}^{n_z}$ . The function  $\mathbf{h} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_z}$  converts system states to causal sensor observations. The function  $\mathbf{h}^{-1} : \mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_x}$  is defined as the inverse function if  $\mathbf{h}^{-1}(\mathbf{h}(\mathbf{x})) = \mathbf{x}$ .  $\mathbf{h}^{-1}(\cdot)$  exists if  $\mathbf{h}(\cdot)$  is a bijection i.e. having a one-to-one mapping between the elements of  $\{\mathbf{x}_k\}$  and  $\{\mathbf{z}_k\}$  as well as surjective [87]. For most observable systems, the sensor function  $\mathbf{h}(\cdot)$  satisfies this constraint [56], but the existence of the inverse function is not guaranteed (e.g. Lorentz system [97]).

Rearranging the sensor equation and using a different notation for inverse function,  $\mathbf{h}^{-1}(\cdot) = \mathbf{g}(\cdot)$ , we have

$$\mathbf{g}(\mathbf{z}_k - \mathbf{v}_k) = \mathbf{x}_k \quad (4.6.2)$$

Using the first order Taylor series expansion about the expected measurement conditioned on posterior estimate available  $\hat{\mathbf{z}}_{k|k} = \mathbb{E}(\mathbf{z}_k | \hat{\mathbf{x}}_{k|k})$ .

$$\mathbf{g}(\mathbf{z}_k - \mathbf{v}_k) \approx \mathbf{g}(\hat{\mathbf{z}}_{k|k}) + \nabla_{\mathbf{g}}^{\hat{\mathbf{z}}_{k|k}} (\mathbf{z}_k - \mathbf{v}_k - \hat{\mathbf{z}}_{k|k}), \quad (4.6.3)$$



where  $\nabla_g^{\hat{\mathbf{z}}_{k|k}}$  is the Jacobian matrix of  $\mathbf{g}(\cdot)$  evaluated at  $\hat{\mathbf{z}}_{k|k}$ . Substituting the first-order linearization in equation (4.6.2) and rearranging the terms, we have,

$$\underbrace{\mathbf{g}(\hat{\mathbf{z}}) + \nabla_g^{\hat{\mathbf{z}}_{k|k}}(\mathbf{z} - \hat{\mathbf{z}})}_{\text{converted measurement}} = \mathbf{x}_k + \underbrace{\nabla_g^{\hat{\mathbf{z}}_{k|k}} \mathbf{v}_k}_{\text{converted meas. noise}} \quad (4.6.4)$$

where the generalized converted measurement covariance  $\mathbf{R}^g$  characterized by the noise vector  $\nabla_g^{\hat{\mathbf{z}}_{k|k}} \mathbf{v}_k$  is given by,

$$\mathbf{R}_k^g = \nabla_g^{\hat{\mathbf{z}}_{k|k}} \mathbf{R}_k \left( \nabla_g^{\hat{\mathbf{z}}_{k|k}} \right)', \quad (4.6.5)$$

where  $\mathbf{R}_k$  is the covariance matrix corresponding to the measurement noise  $\mathbf{v}_k$ . The linearization has been attempted in tracking applications using radar and it was found that the conversion is biased, though  $\mathbf{R}^g$  stays consistent for a maximum azimuth standard deviation of 23° [59].

#### 4.6.2 Second Order Linearization

One can also perform the second order linearization in which the converted measurement is given by (omitting the time index  $k$ ),

$$\mathbf{z}^m = \mathbf{g}(\hat{\mathbf{z}}) + \nabla_g^{\hat{\mathbf{z}}}(\mathbf{z} - \hat{\mathbf{z}}) + \frac{1}{2} \begin{bmatrix} \vdots \\ (\mathbf{z} - \hat{\mathbf{z}})' \mathcal{H}_i^{\hat{\mathbf{z}}} (\mathbf{z} - \hat{\mathbf{z}}) \\ \vdots \end{bmatrix} \quad (4.6.6)$$

where  $\mathcal{H}_i^{\hat{\mathbf{z}}}$  is the Hessian matrix corresponding to the  $i^{\text{th}}$  element of the vector valued function  $\mathbf{g}(\cdot)$  evaluated at  $\hat{\mathbf{z}}$ ,

$$\mathcal{H}_i^{\hat{\mathbf{z}}} = [\nabla\nabla'\mathbf{g}_i(\mathbf{z})] |_{\mathbf{z}=\hat{\mathbf{z}}}. \quad (4.6.7)$$

The element corresponding to the  $i^{\text{th}}$  row and  $j^{\text{th}}$  column of the converted measurement covariance  $\mathbf{R}^g$  is given by (proof in appendix),

$$\mathbf{R}^g = \frac{1}{4} \begin{bmatrix} 2 \operatorname{tr}(\mathcal{H}_1^{\hat{\mathbf{z}}} \mathbf{R} \mathcal{H}_1^{\hat{\mathbf{z}}} \mathbf{R}) & \cdots & 2 \operatorname{tr}(\mathcal{H}_1^{\hat{\mathbf{z}}} \mathbf{R} \mathcal{H}_{n_x}^{\hat{\mathbf{z}}} \mathbf{R}) \\ \vdots & \ddots & \vdots \\ 2 \operatorname{tr}(\mathcal{H}_{n_x}^{\hat{\mathbf{z}}} \mathbf{R} \mathcal{H}_1^{\hat{\mathbf{z}}} \mathbf{R}) & \cdots & 2 \operatorname{tr}(\mathcal{H}_{n_x}^{\hat{\mathbf{z}}} \mathbf{R} \mathcal{H}_{n_x}^{\hat{\mathbf{z}}} \mathbf{R}) \end{bmatrix} + \nabla_{\mathbf{g}}^{\hat{\mathbf{z}}} \mathbf{R} \nabla_{\mathbf{g}}^{\hat{\mathbf{z}T}} \quad (4.6.8)$$

where  $\operatorname{tr}(\cdot)$  refers to the trace operator and  $n_x$  is the dimension of the state vector.

Once the converted measurement model is obtained, the initialization of the state error covariance at FC is a mere substitution, as in equation (4.5.11). After the covariance is initialized, the next step is to recursively calculate the posterior covariance at each fusion instant (on-demand).

## 4.7 Covariance Recursion at Fusion Center

After initialization at the FC, the covariance update requires the following information at  $k$ .

- (i) A vector containing measurement update times from the local node  $\tau_k$ . This

vector contains the timestamps at which measurement update has been performed at the local node since the last fusion time.

- (ii) For non-linear measurement models, the measurement Jacobian matrix (for EKF based recursion). Or,
- (iii) The converted measurement covariance (for CMKF based recursion).

Suppose we have initialized the posterior covariance  $\mathbf{P}_{k-\Delta k|k-\Delta k}^j$  for a local node  $j$ . Then, at time  $k$  at the fusion center, the subsequent recursion for covariance consists of

### Prediction Step

One-step prediction of covariance is a straightforward equation given by,

$$\mathbf{P}_{k|k-\Delta k}^j = \mathbf{F}(\Delta k)\mathbf{P}_{k-\Delta k|k-\Delta k}^j\mathbf{F}(\Delta k)' + \mathbf{Q}(\Delta k) \quad (4.7.1)$$

where  $\Delta k$  is the time difference since the last covariance update (or initialization).  $\mathbf{F}(\cdot)$  and  $\mathbf{Q}(\cdot)$  are the state-transition matrix and the process noise covariance matrix, respectively. For non-linear process models,  $\mathbf{F}(\cdot)$  can be linearized as in the EKF algorithm [14]. In the case of multiple targets, the posterior estimate from a local node would be associated to a target first and then the covariance prediction shall be performed.

## Update Step

The Kalman gain for a linear measurement model is given by,

$$\mathbf{K}_k^j = \mathbf{P}_{k|k-\Delta k}^j \mathbf{H}_k^{j'} \left( \mathbf{H}_k^j \mathbf{P}_{k|k-\Delta k}^j \mathbf{H}_k^{j'} + \mathbf{R}_k^j \right)^{-1} \quad (4.7.2)$$

and the subsequent covariance update

$$\mathbf{P}_{k|k}^j = (\mathbf{I} - \mathbf{K}_k^j \mathbf{H}_k^j) \mathbf{P}_{k|k-\Delta k}^j, \quad (4.7.3)$$

where  $\mathbf{H}_k^j$  is the measurement-state mapping matrix. For the non-linear measurement model (as in equation (4.6.1)), there are various options depending on the computation prowess available at the FC such as:

- The EKF recursion suggests using the measurement Jacobian in place of  $\mathbf{H}_k^j$  evaluated at the posterior state estimate, which is available at the F.C.

$$\mathbf{H}_k^j \approx \Delta \mathbf{h}(\mathbf{x}_k) |_{\hat{\mathbf{x}}_{k|k}}. \quad (4.7.4)$$

- Alternatively, one can also use the CMKF update similar to the description in section 4.4. The  $\mathbf{H}_k^j$  in that case becomes an Identity matrix and the sensor covariance is replaced by the converted measurement covariance  $\mathbf{R}_k^t$  (conditioned on posterior state), or  $\mathbf{R}_k^g$  (first or second-order linearization).
- For more accurate computation, one can also use unscented transformation which requires calculation of deterministic sigma-points [101].

Since the posterior estimate is available at the fusion center, calculation of the

Jacobian  $\Delta\mathbf{h}(\mathbf{x})|_{\hat{\mathbf{x}}_{k|k}}$  or the converted measurement covariance  $\mathbf{R}^g$  is straightforward.

#### 4.7.1 A Look at Measurement Update Times Vector

Ideally, the track covariance recursion does not depend on the track estimate and hence can be performed anywhere given the knowledge of system parameters. We exploit this property to reconstruct the track covariance at the fusion center. A major caveat, however, in the case of non-full communication rate is to know when to update the covariance and when to just predict since the sensor might miss the measurement. Excluding this condition and updating the track covariance each time an estimate is received will lead to an optimistic and possibly inconsistent covariance.

Suppose at time  $k$ , the previous track covariance  $\mathbf{P}_{k-\Delta k|k-\Delta k}^j$  for node  $j$  is available at FC. The local node then sends the posterior track estimate  $\hat{\mathbf{x}}_{k|k}$  along with the measurement update-times  $\tau_k$  which looks like,

$$\tau_k = \begin{bmatrix} t_{k-\Delta k+u_1} & t_{k-\Delta k+u_2} & \cdots & t_k \end{bmatrix}^T \quad (4.7.5)$$

where  $t_{k-\Delta k}$  is the last track-covariance update time at FC and  $t_{k-\Delta k+u_i}$  is the time of  $i^{\text{th}}$  covariance update at local node after  $t_{k-\Delta k}$ . The fusion center then starts the covariance predict-update cycle for each time instant,  $t_{k-\Delta k+u_i}$  by calculating the differences since the last update, until time  $t_k$  is reached. Note that for non-linear systems, covariance update requires knowledge of local state estimate at  $k$  for generating linearized models. This can be approximated by using a predicted estimate conditioned on the latest posterior estimate available.  $\tau_k$  is an undesired variable which is a compromise for not transmitting the local track-covariance.

An efficient alternative is to assume that the sensor sample-time is constant and the local node transmits a vector containing Boolean variables to depict whether prediction or update happened at a particular sample-instant. The sensor sample-time (assumed constant) can then be calculated from the number of elements ( $n$ ) in the vector and used to estimate time-stamp  $k$ .

$$\Delta T = \frac{\Delta k}{n}, \quad (4.7.6)$$

where  $\Delta k$  is the time-difference since last fusion time.

An example in this case is,

$$\tau'_k = \begin{bmatrix} \underbrace{1}_{\text{update at}} & 1 & \underbrace{0}_{\text{predict at}} & \dots \\ (k - \Delta k + \Delta T) & & (k - \Delta k + 3\Delta T) & \end{bmatrix}^T. \quad (4.7.7)$$

which is equivalent to transmitting a scalar variable. Note that the only assumption is that the sensor sample-time is constant throughout  $\Delta k$ .

Alternatively, for known constant local sensor sample-time  $\Delta T$ , we can calculate the number of entries  $n$  in  $\tau'_k$  and its particular time stamp, using the time difference,  $\Delta k$ . Then, we can solve a multi-hypothesis problem to find which hypothesis was assumed by the local node to arrive at the latest estimate.

To understand this, assume that  $\Delta T = 1$  second and  $\Delta k = 3$  seconds, therefore  $\tau'_k$  contains only one entry:

$$\tau'_k = \begin{bmatrix} t_{i=2} \end{bmatrix}, \quad (4.7.8)$$

The estimate and covariance at time  $t_{k-\Delta k}$  is available at FC, while only the estimate is available at  $t_k$ . Denoting prediction by  $p$ , and update by  $u$ , we have four hypotheses to choose from :  $\mathcal{H} = \{(p_i, p_k); (p_i, u_k); (u_i, p_k); (u_i, u_k)\}$ . Where the first entry of each element denotes prediction or update from time  $k - \Delta k$  to  $i$  and the second entry from time  $i$  to  $k$ . Thus, we have a set of covariance matrices at  $k$  corresponding to each entry. Our task is to select the covariance from  $\mathcal{H}$  which minimizes the following K-L divergence:

$$\mathbf{P}_k^* = \arg \min_{\{\mathbf{P} \in \mathcal{H}\}} D_{KL}(\mathbf{P}, \mathbf{R}_k^t) \quad (4.7.9)$$

where,

$$D_{KL}(\mathbf{P}, \mathbf{R}_k^t) = \frac{1}{2} \left\{ \text{tr}(\mathbf{P}^{-1} \mathbf{R}_k^t) - d + \ln \frac{\det(\mathbf{P})}{\det(\mathbf{R}_k^t)} \right\}. \quad (4.7.10)$$

where  $d$  is the dimension of  $\hat{\mathbf{x}}_{k|k}$ . The above equation is simply the K-L divergence between two Gaussian densities with same means and different covariances.

Thus, it is possible to recursively propagate covariance in time without the knowledge of  $\tau'_k$ , but the process is computationally expensive and approximate for non-linear systems (due to non-availability of local estimates). The order of enumeration for cost computing is  $\mathcal{O}(2^{n+1})$  which is not practical for large fusion sample period ( $n$ ).

### 4.7.2 Necessity of $\tau_k$ .

The question regarding the necessity of the measurement update vector times  $\tau_k$ , depends on if the multiple predict-update cycles can be replaced by a single predict-update since sample time is a sensor parameter and not a target parameter. To understand this, the generalization of the discrete-time Riccati equation is revisited.

Assume that the last known updated track covariance  $\mathbf{P}_{k_a|k_a}$  at time-step  $k_a$  and sensor sampling time is  $\Delta T$ . Then, at time  $k_b > k_a$ , for a single update cycle (unknown sample-time), the covariance  $\mathbf{P}_{k_b|k_b}$  is given by,

$$\begin{aligned}\mathbf{P}_{k_b|k_b} &= \bar{\mathbf{K}}_b \left( \mathbf{F}_{(k_b, k_a)} \mathbf{P}_{k_a|k_a} \mathbf{F}'_{(k_b, k_a)} + \mathbf{Q}_{(k_b, k_a)} \right), \\ &= \bar{\mathbf{K}}_{k_b} \left( \mathbf{F}_{(k_b, k_a)} \mathbf{P}_{k_a|k_a} \mathbf{F}'_{(k_b, k_a)} \right) + \bar{\mathbf{K}}_{k_b} \mathbf{Q}_{(k_b, k_a)},\end{aligned}\quad (4.7.11)$$

where,

$$\bar{\mathbf{K}}_{k_b} = (\mathbf{I} - \mathbf{K}_{k_b} \mathbf{H}_{k_b}). \quad (4.7.12)$$

Where  $\mathbf{I}$  is the identity matrix of appropriate dimension,  $\mathbf{F}_{(k_b, k_a)}$  and  $\mathbf{Q}_{(k_b, k_a)}$  are the state transition and process covariance matrices respectively with argument  $k_b - k_a$ , and  $\mathbf{K}_{k_b}$  and  $\mathbf{H}_{k_b}$  are respectively the Kalman gain and the measurement matrix at time  $k_b$ .



However, the exact covariance  $\mathbf{P}_{k_b|k_b}^e$  with the knowledge of  $\Delta T$  is given by,

$$\begin{aligned} \mathbf{P}_{k_b|k_b}^e &= \left( \prod_{i=0}^n \bar{\mathbf{K}}_{n-i} \mathbf{F}_{\Delta T} \right) \mathbf{P}_{k_a|k_a} (\mathbf{F}_{\Delta T}^T)^{n+1} \\ &+ \bar{\mathbf{K}}_{k_b} \left\{ \sum_{j=1}^{n-1} \prod_{l=1}^j \mathbf{F}_{\Delta T} \bar{\mathbf{K}}_{n-l} \mathbf{Q} (\mathbf{F}_{\Delta T}^T)^i \right\} + \bar{\mathbf{K}}_{k_b} \mathbf{Q}_{\Delta T} \end{aligned} \quad (4.7.13)$$

where  $n$  is a positive integer such that  $k_b = k_a + n\Delta T$ . It can be seen that covariance in equation (4.7.13) is considerably lower than the one in equation (4.7.11). The proof of this statement is straightforward given the fact that  $\bar{\mathbf{K}} \preceq \mathbf{I}$  and  $(\mathbf{F}_{\Delta T})^n = \mathbf{F}_{n\Delta T} \succeq \mathbf{F}_{\Delta T}, \forall \{n \geq 1\}$  thus,  $\prod_i^n \bar{\mathbf{K}}_i \mathbf{F}_{\Delta T}^T \preceq \bar{\mathbf{K}}_n \mathbf{F}_{n\Delta T}^T$  and similarly for  $\mathbf{Q}_{\Delta T}$  in the sense of positive-definiteness. This is also illustrated in Fig. 4.2 where Frobenius norm is plotted for fixed  $k_a$  and the final time-step  $k_b$  varying from 0 to 100 seconds. The process noise intensity  $\tilde{q}$  is taken as 0.01 and 1 in Fig. 4.2a and 4.2b respectively with an initial track covariance as  $100\mathbf{I}$  and measurement covariance  $50\mathbf{I}$ .

### 4.7.3 Consistency

In general, for non-linear systems, the linearized Kalman filters do not guarantee consistency. For instance, the standard EKF is a first order approximation which usually remains consistent for small value of noise standard deviations ( $\sigma_\theta \leq 1^\circ$ ) [14], this also applies to the CMKF. Since the local estimates are not guaranteed to be consistent, the reconstructed covariance is also vulnerable to inconsistency. There is, however, a way to *force* the local NEES to be equal to the NEES obtained using reconstruction at each time instant if the local statistic,  $\hat{\mathbf{x}}_{k|k}^T \mathbf{P}_{k|k}^{-1} \hat{\mathbf{x}}_{k|k}$  (scalar) is made available at the FC. In that case, we can calculate the approximate scale factor  $\gamma_k$

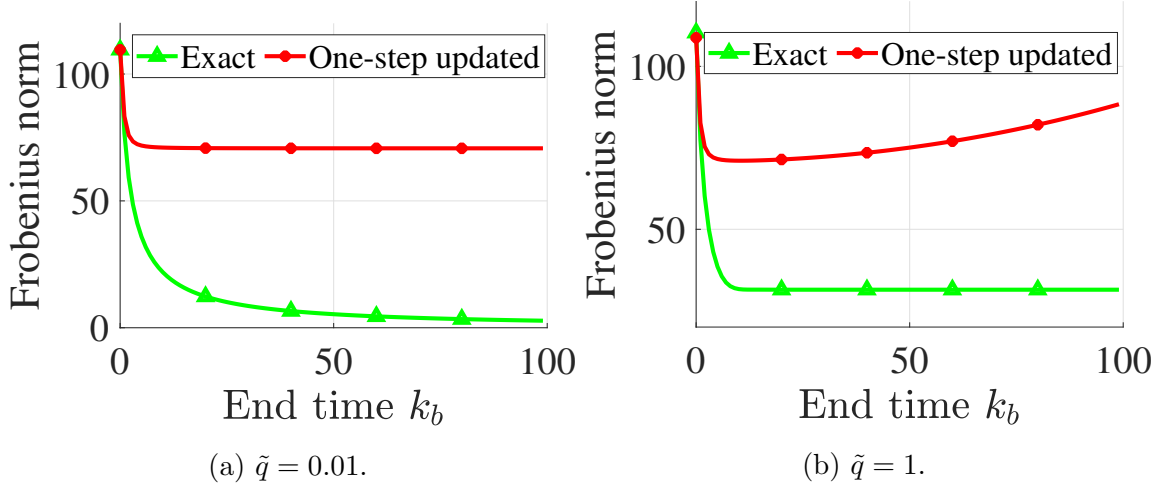


Figure 4.2: Frobenius norm of exact covariance in equation (4.7.13) and one-step covariance equation in (4.7.11) v/s process noise intensity,  $\tilde{q}$  for an arbitrary linear system.

such that  $\mathbf{P}_{k|k} = \gamma_k \mathbf{P}_{k|k}^r$  and  $\gamma_k$  is approximated as,

$$\gamma_k \approx \frac{\hat{\mathbf{x}}_{k|k}^T \mathbf{P}_{k|k}^{r-1} \hat{\mathbf{x}}_{k|k}}{\hat{\mathbf{x}}_{k|k}^T \mathbf{P}_{k|k}^{-1} \hat{\mathbf{x}}_{k|k}}, \quad \hat{\mathbf{x}}_{k|k}^T \mathbf{P}_{k|k}^{-1} \hat{\mathbf{x}}_{k|k} \neq 0. \quad (4.7.14)$$

where  $\mathbf{P}_{k|k}^r$  is the reconstructed covariance at FC and  $\mathbf{P}_{k|k}$  is the posterior covariance at time  $k$  for the same node. It can be seen that  $\gamma_k > 0$  but is not necessarily greater than 1, thus, the reconstructed covariance is not always inflated.

The aforementioned technique is useful especially in the case when the local node is using association techniques to declutter the measurements, like the probability data association tracker (PDA). The posterior covariance associated with the PDA tracker [see equation (4.8.16)] is inflated in comparison to the standard filtering algorithms used for reconstruction due to uncertainty in the measurement origin. Hence, multiplying the reconstructed covariance by the fudge factor  $\gamma_k \geq 1$  from equation (4.7.14) is a suitable option.

## 4.8 Simulation

Two simulation scenarios are presented in this section. The first focuses on evaluating the properties of the reconstructed covariance matrix, and the second scenario focuses on the fusion performance with the reconstructed covariance. In the first scenario, which is a single sensor problem, the sensor propagates local state estimates with a fixed probability, and we compare local covariance with its reconstructed counterpart. The second scenario is that of a multi-target multi-sensor tracking in the presence of clutter. The targets are assumed to be widely separated and thus, the local tracker uses the probability data association filter (PDAF) that takes care of the uniform distributed clutter as well.

### 4.8.1 Performance Metrics

The simulations are evaluated on the following metrics:

- (i) **Root mean-square error (RMSE)**: RMSE evaluates the absolute filtering performance in simulation where the true system trajectory is known. At any time-step  $k$ , the RMSE is given by the equation,

$$\text{RMSE}_k = \sqrt{\frac{1}{M} \sum_{m=1}^M \|\hat{x}_{m,k} - x_k\|^2}, \quad (4.8.1)$$

where  $M$  is the number of Monte-Carlo (MC) iterations over which the root-mean is evaluated.  $\hat{x}_{m,k}$  is the filter estimate at  $k^{\text{th}}$  time-step of  $m^{\text{th}}$  MC run. Note that the truth  $x_k$  is the same throughout all MC iterations.

- (ii) **Average normalized estimation error squared (ANEES)**: ANEES is a

measure of estimator consistency, which is based on the fact that a  $\chi^2$  distribution has a mean equal to its degree of freedom. Thus, the following expression

$$\text{ANEES}_k = \frac{1}{M} \left[ (\hat{\mathbf{x}}_k - \mathbf{x}_k)^T \mathbf{P}_{k|k}^{-1} (\hat{\mathbf{x}}_k - \mathbf{x}_k) \right] \quad (4.8.2)$$

should have  $n_x$  degrees of freedom (asymptotically). Since the number of MC runs are finite, the ANEES for a consistent estimator should lie between the appropriate confidence bounds given by the tail probabilities of respective  $\chi^2$  density [14].

- (iii) **Mean deviation:** Relative error of the actual and reconstructed covariance matrices can be compared without the knowledge of truth using the mean deviation over  $M$  Monte-Carlo runs. For a covariance matrix  $\mathbf{P}_{k|k}$  at time-step  $k$ , the mean deviation for the scalar variable  $x$  is

$$\text{MD} = \frac{1}{M} \sum_{m=1}^M \sqrt{\text{cov}(x)}. \quad (4.8.3)$$

We extend this to target tracking scenarios in which the mean deviation of position and velocities are computed as (in 2D),

$$\text{MD}_{\text{pos}} = \frac{1}{M} \sum_{m=1}^M \sqrt{\text{cov}(x) + \text{cov}(y)} \quad (4.8.4)$$

$$\text{MD}_{\text{vel}} = \frac{1}{M} \sum_{m=1}^M \sqrt{\text{cov}(\dot{x}) + \text{cov}(\dot{y})} \quad (4.8.5)$$

- (iv) **Frobenius distance between matrices:** Based on the Frobenius norm [46],

the *Frobenius distance* between the two matrices  $\mathbf{A}$  and  $\mathbf{B}$  is given as

$$F_{\mathbf{A},\mathbf{B}} = \sqrt{\text{tr} [(\mathbf{A} - \mathbf{B})(\mathbf{A} - \mathbf{B})']}, \quad (4.8.6)$$

where  $\text{tr}[\cdot]$  refers to the trace operator. Removing the  $\text{tr}[\cdot]$  operator, equation (4.8.6) can also be written as,

$$F_{\mathbf{A},\mathbf{B}} = \sqrt{\sum_{i=1}^{n_x} \sum_{j=1}^{n_x} (a_{ij} - b_{ij})^2}, \quad (4.8.7)$$

where  $a_{ij}$  and  $b_{ij}$  are respectively, the corresponding elements of  $n_x \times n_x$  sized matrices  $\mathbf{A}$  and  $\mathbf{B}$ . Thus,  $F_{\mathbf{A},\mathbf{B}}$  can be thought of as an  $\mathcal{L}_2$  norm which tends to 0 as the individual elements  $a_{ij}$  and  $b_{ij}$  tend to be equal to each other. In our case, the mean of this operator is calculated over all Monte-Carlo runs. Thus,

$$\hat{\mathbf{F}} = \frac{1}{M} \sum_{m=1}^M F_{(\mathbf{P}_{k|k}, \mathbf{P}_{k|k}^r)_m} \quad (4.8.8)$$

where  $\mathbf{P}_{k|k}$  and  $\mathbf{P}_{k|k}^r$  are the actual and reconstructed covariance matrices respectively at time-step  $k$  of  $m^{\text{th}}$  MC run .

### 4.8.2 Single Sensor Covariance Reconstruction.

To compare the reconstructed and original covariance qualitatively, a single sensor simulation is set up to track a nearly constant velocity (NCV) target. In this scenario, the local node sends the posterior track estimates to the FC, wherein the covariance is reconstructed using the procedure explained in Section 4.5 and Section 4.7. The

comparison is based on the metrics explained in subsection 4.8.1 – NEES, Frobenius distance and mean deviation. Note that this scenario is specifically constructed to compare the actual and reconstructed covariance, without any emphasis on the tracking error. This is due to the fact that the reconstructed covariance at FC closely follows the actual covariance at local node, and hence could diverge if the modeling or system parameters are inappropriate.

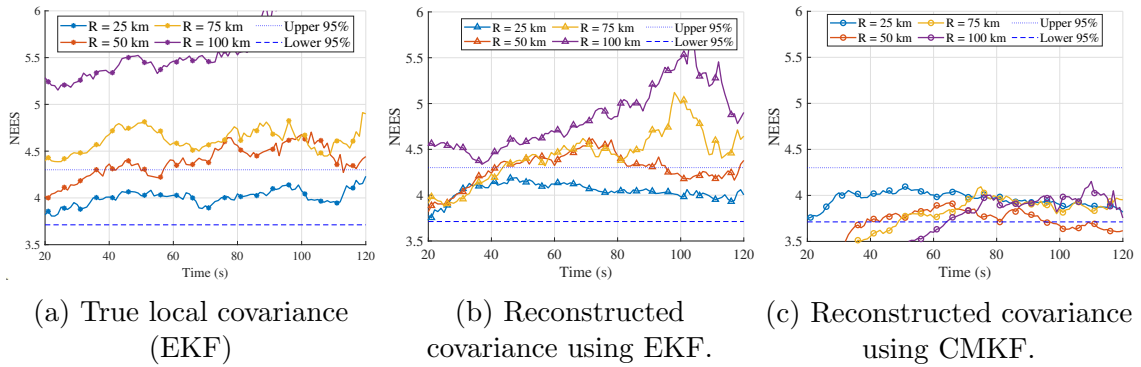


Figure 4.3: True v/s reconstructed NEES for different values of initial range  $R$ , when local sensor is running EKF.

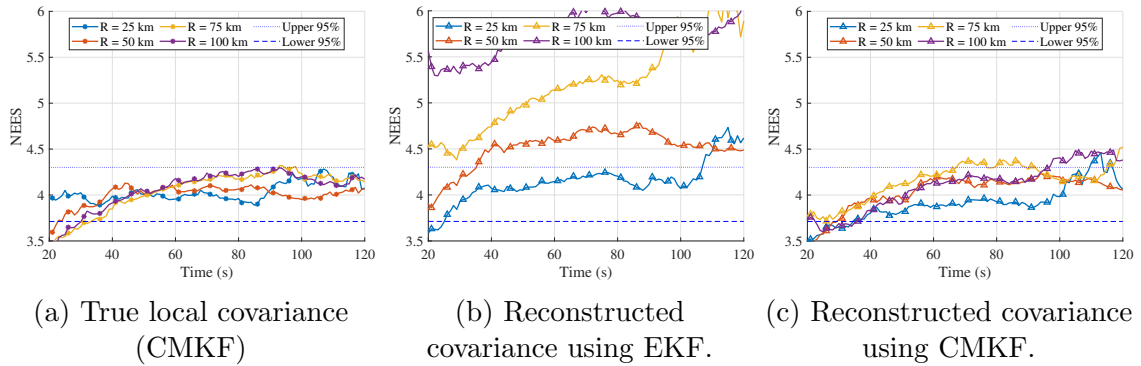


Figure 4.4: True v/s reconstructed NEES for different values of initial range  $R$ , when local sensor is running CMKF.

It is assumed that the local nodes transmit the updated estimate at full communication rate [94]. To emulate real world conditions, the reconstruction happens

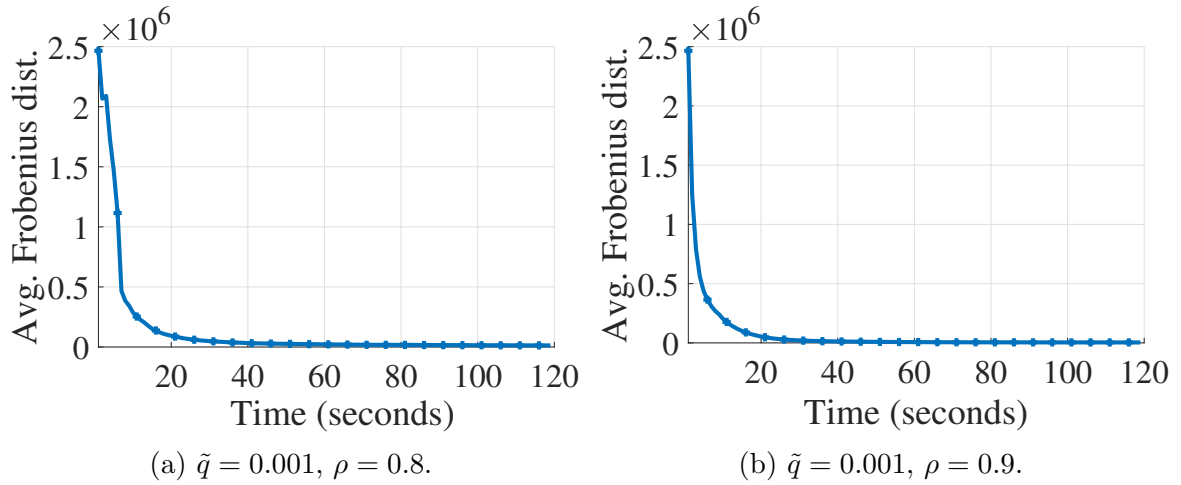


Figure 4.5: Average Frobenius Distance v/s time.

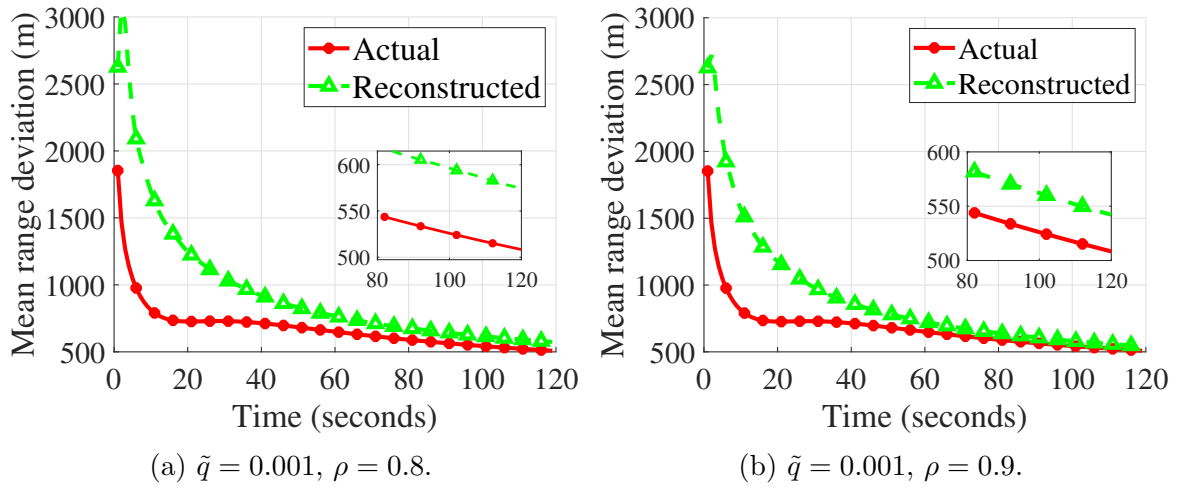


Figure 4.6: Mean deviation of range.

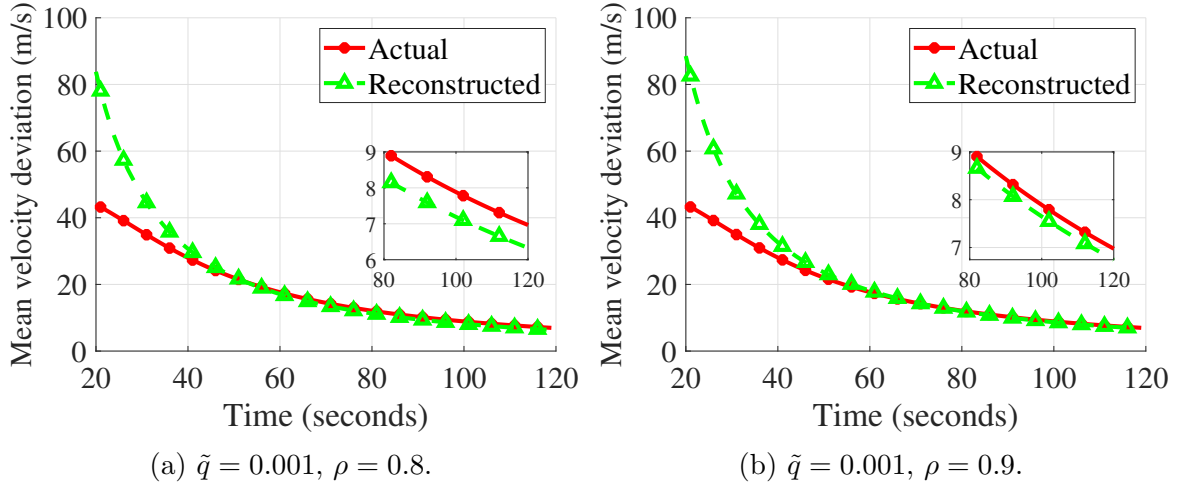


Figure 4.7: Mean deviation of velocity.

with a probability  $\rho$  where  $1 - \rho$  gives the probability of missing estimate at FC. Hence, assuming the local node updates the covariance at each sampling instant with 100% probability, the FC updates the covariance with a probability  $\rho$ . The simulation parameters are listed in Table 4.2.

The NEES for 1000 Monte-Carlo runs is plotted in Fig. 4.3 and Fig. 4.4 for initial range  $R = 25, 50, 75$  and 100 km; process noise intensity,  $\tilde{q} = 0.001$ ; and  $\rho = 0.9$ . Generally, the non-linear filtering method employed at the local node would not be known to the FC, hence NEES is plotted using both the CMKF and EKF framework (see Section 4.7). The case where the local node is running EKF is shown in Fig. 4.3 with the actual NEES in Fig. 4.3a. Note that the local EKF itself is mostly inconsistent for the given values of sensor standard covariance and initial range, this is due to the errors accumulated in the first-order linearization at large range. It can be seen that if the sensor noise standard deviation is reduced ( $\sigma_\theta < 1^\circ$ ), the estimate can be made consistent. The NEES for reconstructed covariance using the EKF framework is shown in Fig. 4.3b and using CMKF framework is plotted in Fig.



4.3c. Though the reconstruction using the CMKF framework is relatively consistent, the EKF framework is more likely to match the original NEES as expected. Even though inconsistent, the EKF framework reconstruction performs better than the original because the sensor Jacobian matrix at FC is constructed with the knowledge of posterior estimate rather than prior estimate, which was the case at the local node. Thereby, the covariance at FC is similar to that of a two-step re-linearized iterated EKF [14] which performs better than the original covariance.

For the case where the local node is running CMKF, the NEES is plotted in Fig. 4.4 with that of the local covariance in Fig. 4.4a. The de-biased version of CMKF given in [21] is employed, which is consistent even at high range and large sensor errors. The EKF reconstruction is shown in Fig. 4.4b which lacks consistency except when the initial range is small. Since the decorrelated unbiased converted measurement (DUCM) estimate is accurate upto third-order linearization [20], a first order EKF cannot outperform its result. One may suggest the use of an iterated EKF at FC, which will also increase computational cost significantly. In Fig. 4.4c, the NEES for reconstruction using CMKF framework is plotted which shows 3–4 points out of 120 to be outside the confidence interval, which is acceptable but still relatively optimistic. The reason for this is that the converted measurement matrix was approximate, since the FC had no information regarding  $\sigma_{r_t}$  and  $\sigma_{\theta_t}$  (see equation (4.5.4)). This simulation suggests that the use of a CMKF based framework proves to be relatively consistent if the F.C has no information regarding the filtering algorithm at the local nodes.

Next, the mean Frobenius distance is plotted for 1000 Monte-Carlo runs in Fig. 4.5. The value of initial range,  $R$  is 75 km and the maximum velocity is 80 m/s.

Fig. 4.5a shows the case with  $\tilde{q} = 0.01$  and  $\rho = 0.8$ , wherein the Frobenius distance converges at the end of simulation after starting at a large value. Since the elements of the covariance matrix are squared error, the Frobenius distance at instant  $k$  is the mean-squared-difference between the elements of  $\mathbf{P}_{k|k}$  and  $\mathbf{P}_{k|k}^r$ . The terminal  $\hat{\mathbf{F}}$  corresponding to this case is  $3.3 \times 10^5$  whereas for the case with  $\tilde{q} = 0.001$  and  $\rho = 0.9$  is  $1.6 \times 10^5$ . Given that the elements of the covariance matrices at such range and sensor noise are of the order of  $10^6$ , the terminal Frobenius distance suggests an error of roughly 10% which is acceptable.

The mean deviation in position and velocity as defined in equation (4.8.4) and (4.8.5) are plotted in Fig. 4.6 and Fig. 4.7 respectively for both the values of  $\rho$ , and  $\tilde{q} = 0.001$ . Both plots show that the reconstructed covariance proves to be a viable alternative, with the terminal difference of  $\sim 50$  meters in range and  $\sim 1$  m/s in velocity for the case with  $\rho = 0.8$ . The numbers tend to improve as the update probability tends to 1 as expected.

### 4.8.3 Three Sensor T2TF Without Covariance

In order to evaluate the robustness of the proposed reconstruction strategy, the fusion result was tested on a real-life 2-D target tracking scenario. The scenario consists of 20 targets which are tracked by three radars within a significant amount of clutter. The scenario is shown in Fig. 4.8a wherein the numbers indicated at the start of each target represents its ID. The sensors have a limited field of view (FOV) which is plotted using circular markers around each sensor.

The parameters used in tracking this scenario are listed in Table 4.3. Note that a nearly constant velocity model (NCV) is assumed for all the tracks, however it is not

Table 4.2: Scenario 1 – Simulation parameters

Parameter	Value
Process noise intensity ( $\tilde{q}$ )	$1E^{-3}, 1E^{-2} \text{ m}^2/\text{s}^3$
Number of targets	1
Detection probability	0.99
Maximum target speed	80 m/s
Sensor std. deviation - range ( $\sigma_r$ )	100 m
Sensor std. deviation - azimuth ( $\sigma_\theta$ )	$4^\circ$
Total simulation time	120 s
Sensor sample time ( $\Delta T$ )	1 s
Packet miss probability ( $\rho$ )	0.9, 0.8

necessarily followed by all target trajectories which might lead to track breakages. The geodetic coordinates according to the WGS84 datum [57] are also provided in Table 3.3 which can be used to replicate the scenario. The targets move in an area spanning nearly  $3600 \text{ km}^2$  whereas the sensors are static at the mentioned coordinates. In Fig. 4.9, the clutter and true measurements are contrasted, which show the intensity of clutter in the scenario.

Since the FOV of each sensor (rotating mono-static radar) is limited as mentioned in Table 4.3, each sensor receives detection for only 12–13 targets out of 20 as depicted in Fig. 4.9. The results also show in Fig. 4.8b that only 19 out of 20 targets could be tracked in such a configuration.

Next, the theory of filtering, association and fusing algorithms used in local nodes and the FC is revisited in this scenario.

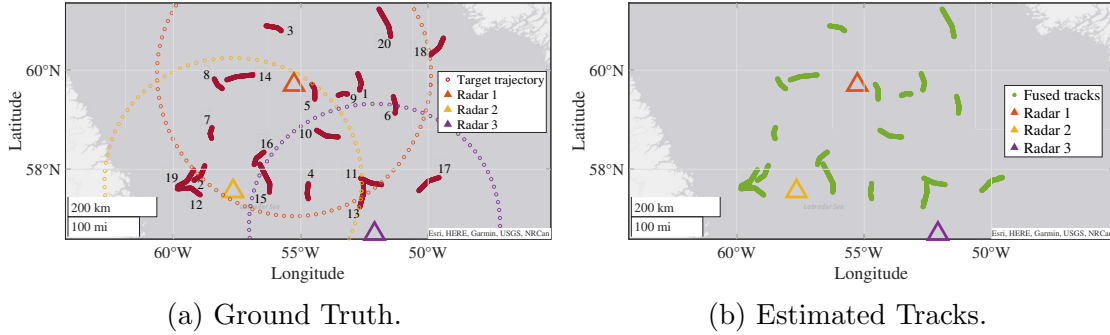


Figure 4.8: Scenario 2 – Comparison of truth and estimated tracks.

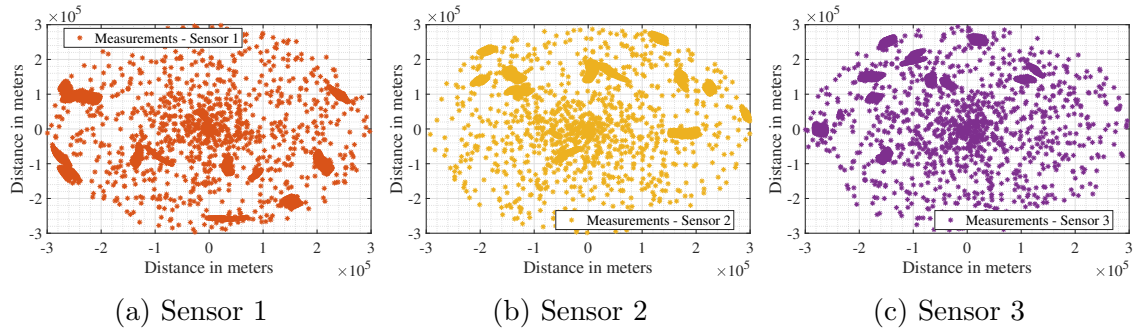


Figure 4.9: Scenario 2 – Measurements v/s Clutter.

### Model Equations

Although the stated coordinates represent a spherical system, the trackers are implemented in a standard mixed coordinate system wherein the process models is Cartesian and the sensor model is spherical in two dimensions. For conversion between the geodetic and local tangent plane coordinates, readers are referred to [28].

The target motion and the measurements respectively, are modeled by the following equations (dropping target and measurement indices),

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{w}_k, \tag{4.8.9a}$$

$$\mathbf{z}_k = \mathbf{h}(x_k, y_k) + \mathbf{v}_k, \tag{4.8.9b}$$

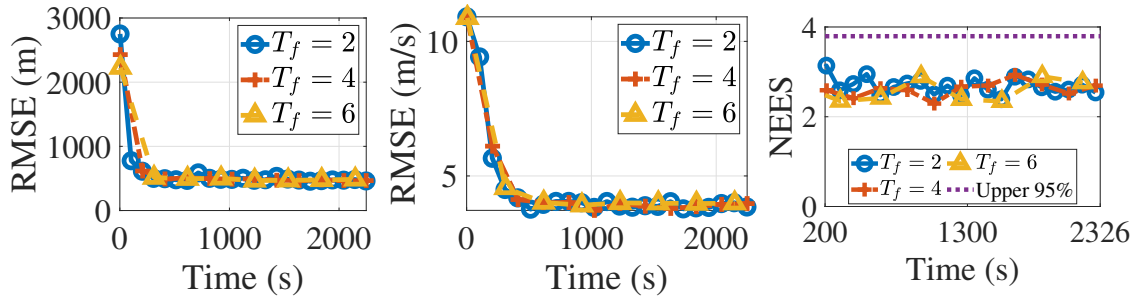


Figure 4.10: Scenario 2 – Target ID 1.

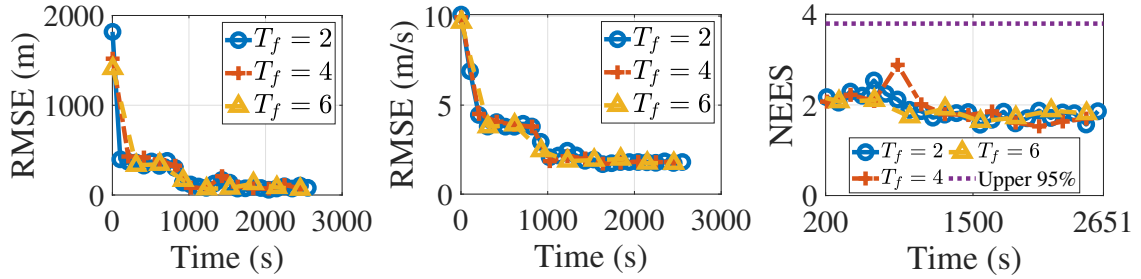


Figure 4.11: Scenario 2 – Target ID 2.

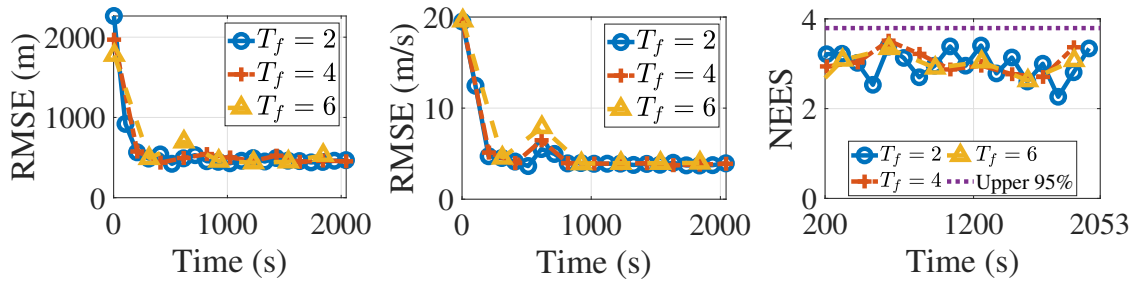


Figure 4.12: Scenario 2 – Target ID 3.

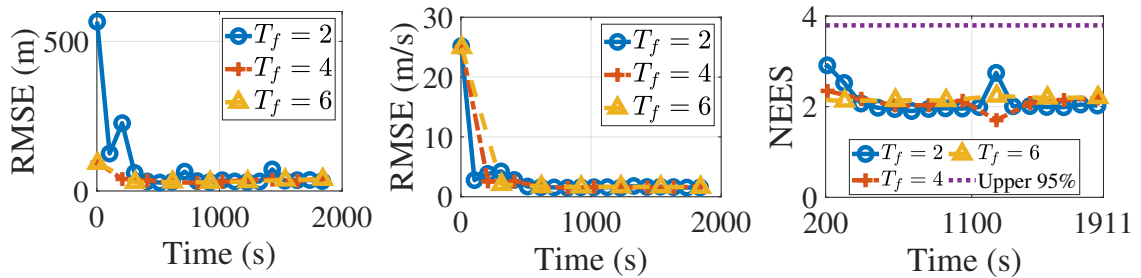


Figure 4.13: Scenario 2 – Target ID 4.

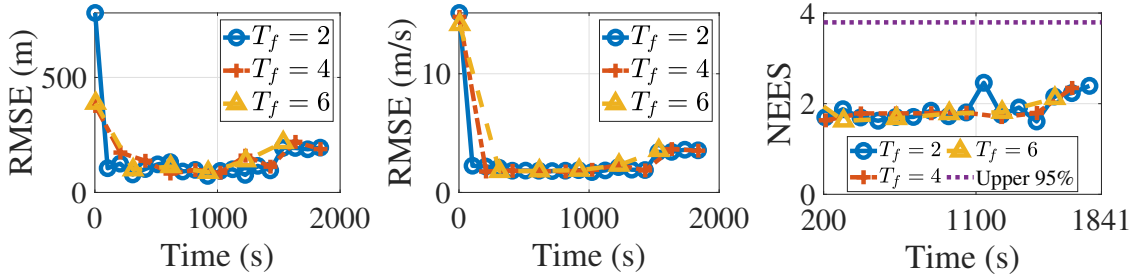


Figure 4.14: Scenario 2 – Target ID 5.

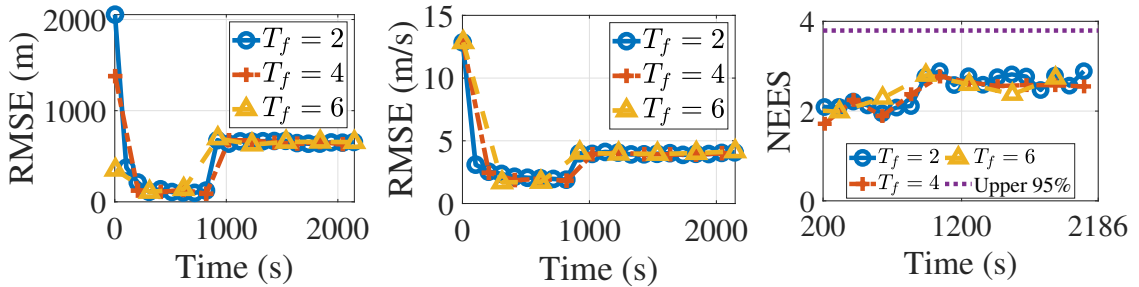


Figure 4.15: Scenario 2 – Target ID 6.

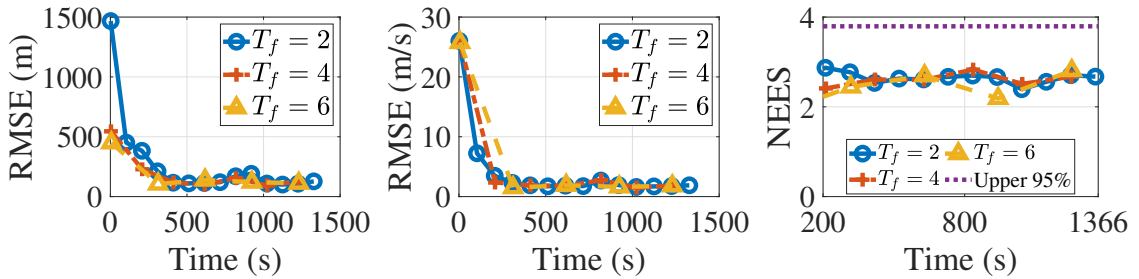


Figure 4.16: Scenario 2 – Target ID 7.

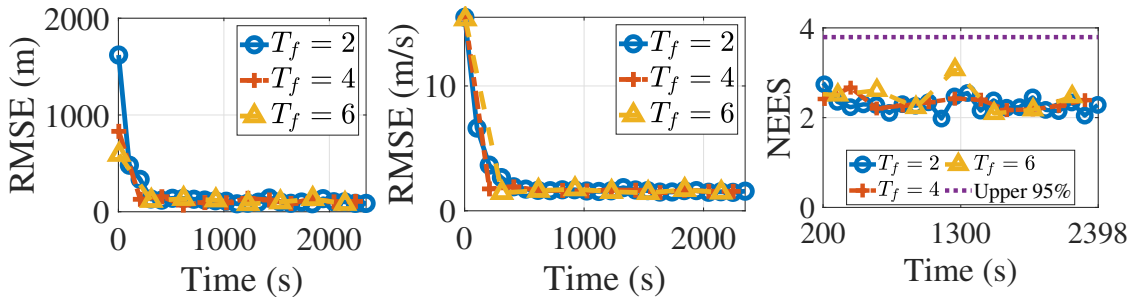


Figure 4.17: Scenario 2 – Target ID 8.

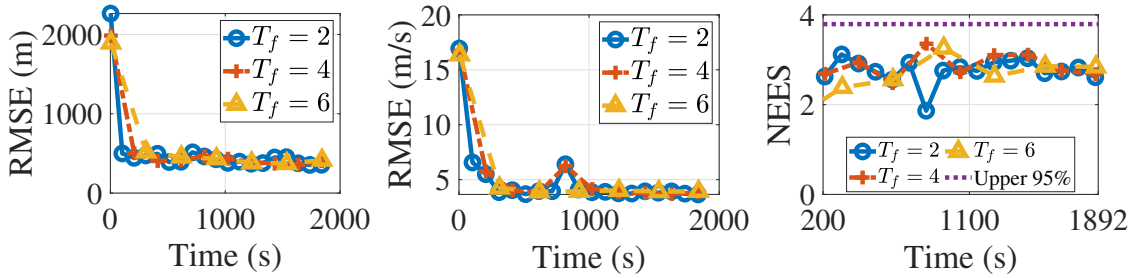


Figure 4.18: Scenario 2 – Target ID 9.

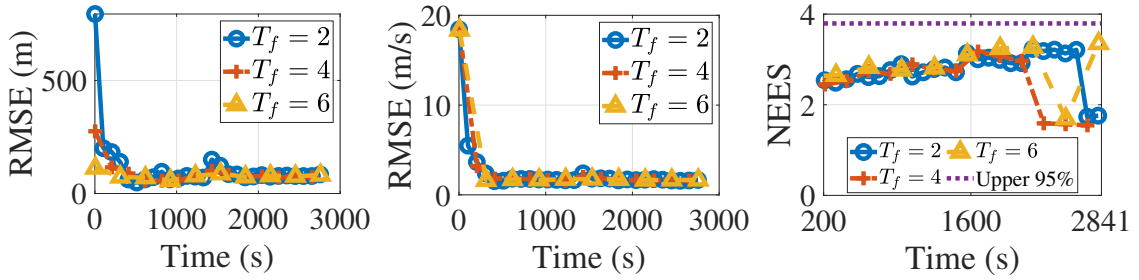


Figure 4.19: Scenario 2 – Target ID 10.

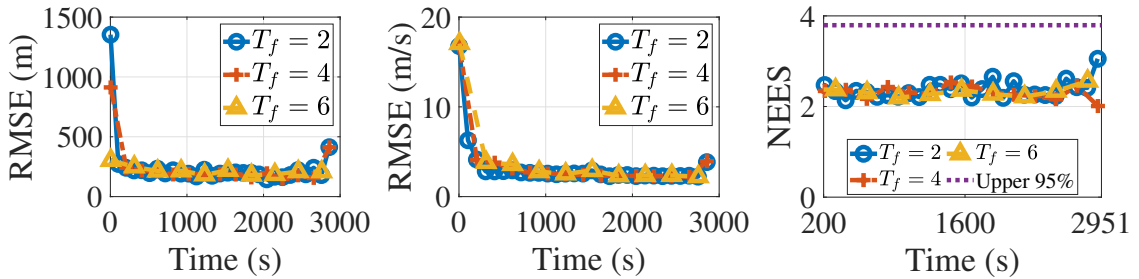


Figure 4.20: Scenario 2 – Target ID 11.

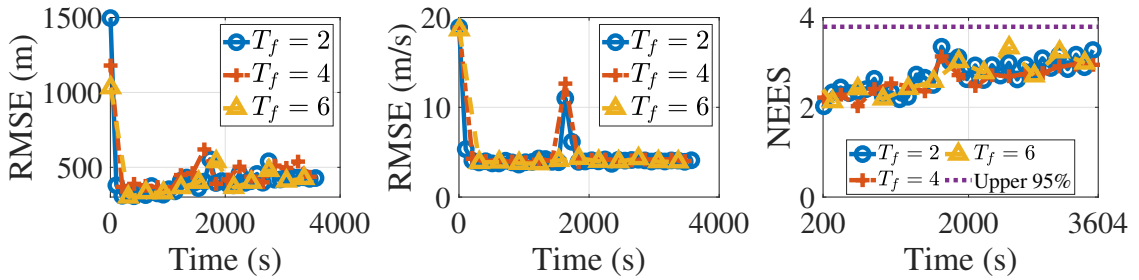


Figure 4.21: Scenario 2 – Target ID 12.

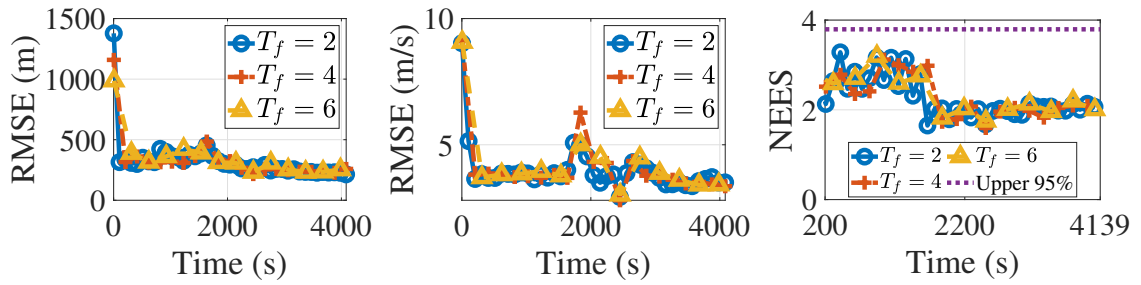


Figure 4.22: Scenario 2 – Target ID 13.

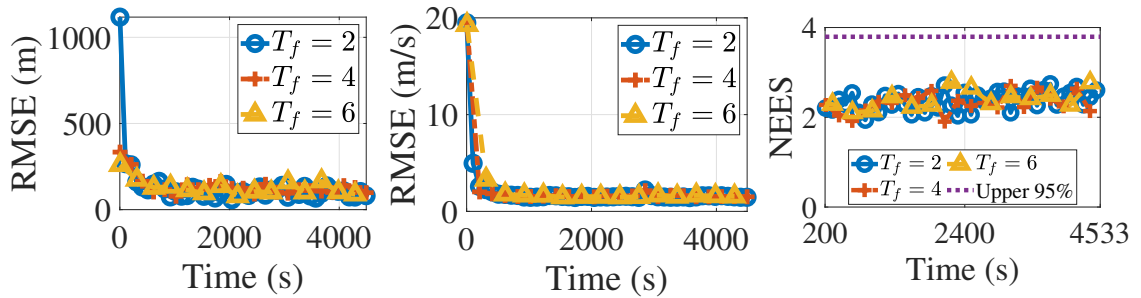


Figure 4.23: Scenario 2 – Target ID 14.

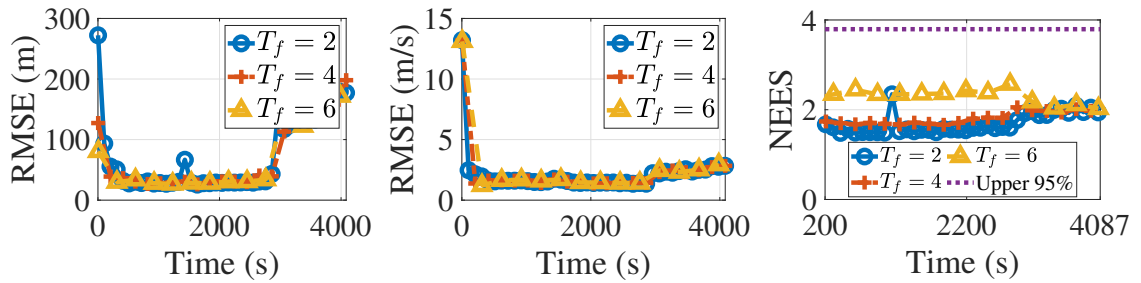


Figure 4.24: Scenario 2 – Target ID 15.

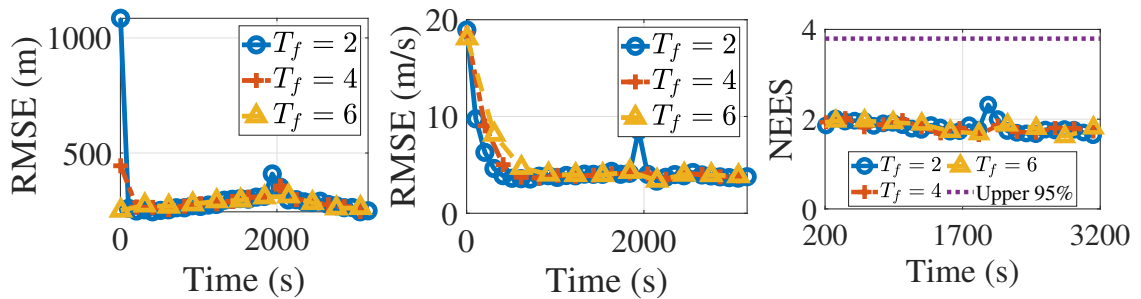


Figure 4.25: Scenario 2 – Target ID 16.



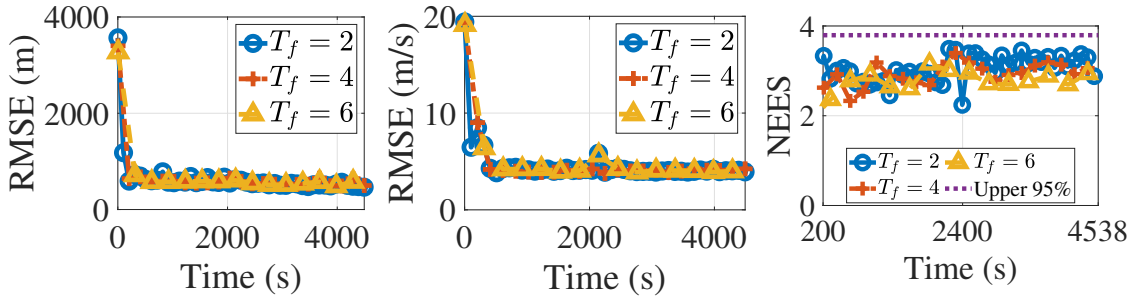


Figure 4.26: Scenario 2 – Target ID 17.

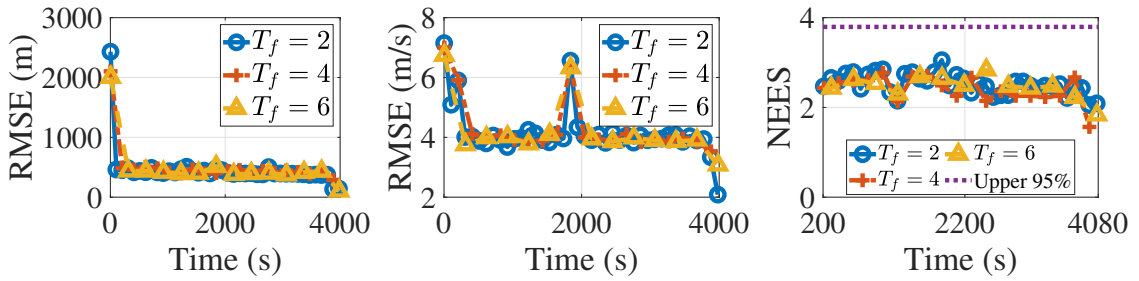


Figure 4.27: Scenario 2 – Target ID 19.

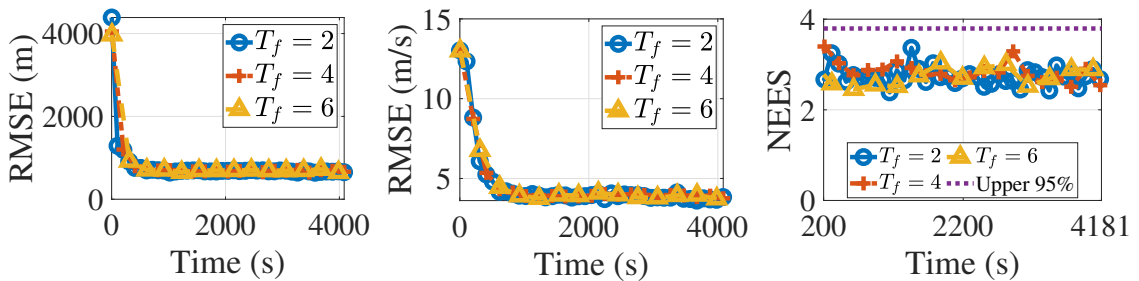


Figure 4.28: Scenario 2 – Target ID 20.

where (using  $\mathbf{I}_n$  for n dimensional identity matrix),

$$\mathbf{F} = \begin{bmatrix} \mathbf{I}_2 & \Delta T \mathbf{I}_2 \\ \mathbf{0}_2 & \mathbf{I}_2 \end{bmatrix} \quad \text{and, } \mathbf{h}(x_k, y_k, z_k) = \begin{bmatrix} r_k & \theta_k \end{bmatrix}^T,$$

Table 4.3: Simulation parameters – Scenario 2

Parameter	Value
Process noise intensity ( $\tilde{q}$ )	0.1 m <sup>2</sup> /s <sup>3</sup>
Number of targets	20
Detection probability	0.99
False alarm probability $P_{fa}$	10 <sup>-6</sup>
Clutter density ( $\lambda$ )	10 <sup>-6</sup>
Maximum target speed	30 m/s
Number of sensors	3
Sensor type	Monostatic 2-D Radar
Sensor coverage	300,000 m × 360°
Sensor std. deviation - range ( $\sigma_r$ )	50 m
Sensor std. deviation - azimuth ( $\sigma_\theta$ )	2°
Gating threshold ( $\mu$ )	0.95
Track deletion threshold (num. of misses)	6
Track loss threshold	500 m
Fusion sample period ( $T_f$ )	4
Total simulation time	4537 seconds
Sensor sample time	1 second

where,

$$r_k = \sqrt{x_k^2 + y_k^2}; \quad \theta_k = \tan^{-1} \left( \frac{y_k}{x_k} \right)$$

$\Delta T$  is the sampling time, and  $\mathbf{w}_k, \mathbf{v}_k$  are uncorrelated, zero-mean Gaussian distributed process noise vector and measurement noise vector respectively, such that  $\forall \{k, j\}$ ,

$$E[\mathbf{w}_k \mathbf{w}_j] = \delta_{kj} \mathbf{Q}; \quad E[\mathbf{w}_k \mathbf{v}_j] = \mathbf{0}; \quad E[\mathbf{v}_k \mathbf{v}_j] = \delta_{kj} \mathbf{R},$$

where  $\mathbf{Q}$  and  $\mathbf{R}$  are the process noise and measurement noise covariance matrix respectively,

$$\mathbf{Q} = \tilde{q} \begin{bmatrix} \frac{\Delta T^3}{3} \mathbf{I}_2 & \frac{\Delta T^2}{2} \mathbf{I}_2 \\ \frac{\Delta T^2}{2} \mathbf{I}_2 & \Delta T \mathbf{I}_2 \end{bmatrix}, \quad \mathbf{R} = \text{diag}(\sigma_r^2, \sigma_\theta^2),$$

and  $\tilde{q}$  is the process noise intensity in meter<sup>2</sup>/second<sup>3</sup>.

### Local Tracking Algorithm

At the local nodes, sensors employ a Probability Data Association (PDA) algorithm with the assumption that the targets are widely separated, which is true in this scenario. The wide separation is quantified by the fact that while validation-gating is performed (see equation (4.8.10)) [12], no two targets can share their associated measurements. Or in other words, the gates do not intersect.

At the crux of the PDA algorithm is the validation gate, which removes all measurement which do not lie in the following probability ellipse [12],

$$(\mathbf{z}_i(k) - \hat{\mathbf{z}}(k|k-1))^T \mathbf{S}^{-1}(k) (\mathbf{z}_i(k) - \hat{\mathbf{z}}(k|k-1)) < \gamma. \quad (4.8.10)$$

Where  $\mathbf{z}_i(k)$  is the  $i^{\text{th}}$  measurement at time  $k$ ,  $\hat{\mathbf{z}}(k|k-1)$  is the one-step predicted measurement at time  $k$ ,  $\mathbf{S}(k)$  is the predicted measurement covariance and  $\gamma$  is obtained from the tables of chi-square distribution for a given confidence level. Then, for each measurement  $\mathbf{z}_i(k)$  in the validated measurements set  $m(k)$ , an association probability  $\beta_i(k)$  is calculated,

$$\beta_i(k) = \begin{cases} \frac{\mathcal{L}_i(k)}{1 - P_D P_G + \sum_{j=1}^{m(k)} \mathcal{L}_i(k)} & i \in 1 : m(k) \\ \frac{1 - P_D P_G}{1 - P_D P_G + \sum_{j=1}^{m(k)} \mathcal{L}_i(k)} & i = 0. \end{cases} \quad (4.8.11)$$

Where  $P_D$ ,  $P_G$  are the detection probability and gate probability respectively.  $\mathcal{L}_i(k)$  is the measurement likelihood ratio, given by,

$$\mathcal{L}_i(k) = \frac{\mathcal{N}[\mathbf{z}_i(k), \hat{\mathbf{z}}(k|k-1), \mathbf{S}(k)]}{\lambda} \quad (4.8.12)$$

$\lambda$  is the clutter intensity and  $\mathcal{N}$  represents the normal distribution. Given the prior estimate  $\hat{\mathbf{x}}(k-1|k-1); \mathbf{P}(k-1|k-1)$ , the filter equations are then (dropping time indices for brevity),

- **Predict :**

$$\hat{\mathbf{x}}(k|k-1) = \mathbf{F}\hat{\mathbf{x}}(k-1|k-1) \quad (4.8.13)$$

$$\mathbf{P}(k|k-1) = \mathbf{F}\mathbf{P}(k-1|k-1)\mathbf{F}^T + \mathbf{Q} \quad (4.8.14)$$

- **Update :**

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{K}(k)\mathbf{v}(k) \quad (4.8.15)$$

$$\mathbf{P}(k|k) = \beta_0\mathbf{P}(k|k-1) + (1 - \beta_0)\mathbf{P}^c(k) + \tilde{\mathbf{P}}(k). \quad (4.8.16)$$

where  $\mathbf{K}(k)$  is the Kalman gain and,

$$\mathbf{P}^c(k) = \mathbf{P}(k|k-1) - \mathbf{K}(k)\mathbf{S}(k)\mathbf{K}(k)^T, \quad (4.8.17)$$

the spread of innovation term,  $\tilde{\mathbf{P}}(k)$  is given by,

$$\tilde{\mathbf{P}}(k) \triangleq \mathbf{K}(k) \left[ \sum_{j=1}^{m(k)} \beta_j(k)\mathbf{v}_j(k)\mathbf{v}_j(k)^T - \mathbf{v}(k)\mathbf{v}(k)^T \right] \mathbf{K}(k)^T. \quad (4.8.18)$$

Here  $\mathbf{v}_i(k)$  is the innovation corresponding to  $i^{\text{th}}$  measurement;  $\mathbf{v}(k)$  is the weighted innovation,  $\mathbf{v}(k) = \sum_i \beta_i(k)\mathbf{v}_i(k)$  and  $\beta(k)$  is calculated in equation (4.8.11).

Since this is a non-linear problem, the CMKF recursion is coupled with PDA in equation (4.8.17) to calculate Kalman gain for linearized measurements. Thus,

$$\mathbf{K}(k) = \mathbf{P}(k|k-1)\mathbf{H}^T (\mathbf{H}\mathbf{P}(k|k-1)\mathbf{H} + \mathbf{R}_k^t)^{-1}, \quad (4.8.19)$$

where  $\mathbf{R}_k^t$  is mentioned in equation (4.5.2) and,

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \quad (4.8.20)$$

### Covariance Reconstruction

The local nodes send their track estimates along with the measurement update times (see equation (4.8.1)). Every time a new track is received at the FC whose covariance is not available, the covariance is initialized using the framework explained in Section 4.5. Then for the subsequent time, EKF or CMKF recursion is employed to update/predict the last covariance based on the measurement update times. For this operation, the FC is assumed to possess sufficient memory, in order to store the covariance for each track.

For this simulation, the fusion sample period of 2, 4 and 6 were used (see Table 4.3) which is the value of sampling instants between fusion demand. It was assumed that the sampling time is constant, therefore, the measurement update times  $\tau_k$  is sent as a binary vector signifying 1 for update and 0 for predict.

To maintain the consistency, the CMKF based covariance recursion is used at the FC.

### Track Association and Fusion

The tracks transmitted by the local nodes are required to be associated after the covariance is reconstructed. In the view of widely separated targets, the association is kept simple for faster execution and global nearest neighbor (GNN) is employed, in

which the assignment is readily made when a single track is associated with the global track. In the event of a conflict, where multiple tracks may appear in the vicinity of a single gate, the suboptimal nearest neighbor approach was used. In this approach, the track with the minimum statistical distance (see equation (4.8.10) ) is chosen.

The tracks associated together are fused using the covariance intersection algorithm, which is described in equation (4.3.8). The fusion factor  $\omega$  is selected by optimizing the following cost criteria

$$\omega^* = \arg \min_{\omega} \text{tr} \left[ \left( \omega \Gamma_{k|k}^{i-1} + (1 - \omega) \Gamma_{k|k}^{j-1} \right)^{-1} \right] \quad (4.8.21)$$

where the  $\text{tr}[\cdot]$  refers to the trace of the resulting matrix and  $\Gamma^i, \Gamma^j$  are the covariance of tracks to be fused at time  $k$ .

## Results

The results, including the metrics position RMSE, velocity RMSE and the NEES over 500 MC iterations, are plotted in Fig. 4.10 – Fig. 4.28 for target IDs 1–20 excluding target 18 which was outside FOV of either sensors. Though generalized-optimal sub-pattern assignment (GOSPA) [79] is the common metric choice for multi-target scenarios, individual metrics were chosen for a better analysis.

Each metric is plotted for different values of the fusion sample period  $T_f$  viz. 2, 4 and 6 seconds. This shows that the results are independent of the fusion period, which is a basic property of T2TF. Having a closer look at the RMSE results, the reader would find a transition in the plots for a specific target. The reason for this lies in the scenario (see Fig. 4.8a) wherein the FOV of each sensor is plotted. For instance, target 17 lies in the FOV of only Radar 3, hence its track estimates never

undergo fusion, resulting in a higher RMSE. On the contrary, targets 4, 10 and 11 lie in the FoV of all three sensors and hence their track estimate are the result of fusion of three local nodes. Similarly, there are targets which cross over the FOV of either sensors, like target 15 which initially stays in FOV of all three sensors and later on terminates in the region covered by Radar 1 and Radar 2. Due to this cross-over, the RMSE displays transition in the later part of the plot. Individual spikes in the RMSE plots could be the result of track breakage, which are not shown in this work for brevity. This is due to the fact the targets do not follow the NCV motion model, which is used in tracking.

The NEES is also plotted in Fig. 4.10—Fig. 4.28. Here also, the transition is visible, the reason is the nature of fusion methodology used i.e. – covariance intersection. C.I. has been proven to be conservative in nature. An estimate pair  $(\hat{\mathbf{x}}_k, \Gamma_k)$  is termed conservative if its covariance is greater than the true mean square error (MSE) [4]. Thus, mathematically for a conservative covariance  $\Gamma_k$ ,

$$\Gamma_k \succeq E [(\hat{\mathbf{x}}_k - \mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T], \quad (4.8.22)$$

where  $\mathbf{x}_k$  is the truth and  $\succeq$  implies greater in the sense of positive-definiteness. Hence, the fusion produces a conservative result due to which the NEES decreases. For instance, the estimate resulting from the fusion of three nodes will have a higher (or in marginal case, equal) NEES if compared with that of the fusion of the two local nodes. Since the covariance of the local track will always be higher (or equal) than the fused covariance, the corresponding NEES will always be lower than the latter. For example, see Fig. 4.15 for target ID-6, which initially lies in the FOV of Radar 1 but later on moves in the vicinity of both Radar 1 and Radar 3. The NEES



is initially low due to local node covariance being higher, then, as the target crosses over to the vicinity of Radar 1 and Radar 3, the NEES increases as the covariance is decreased due to fusion while still being conservative. One-sided 95% confidence level is also constructed in each NEES plot for assessing the consistency. None of the samples for any target is above the 95% confidence, proving the consistency of covariance intersection.

## 4.9 Conclusion

In this chapter, the problem of missing covariance in track fusion architectures was investigated. The proposed method is exact and relies on the fact that covariance recursion in a Kalman filter can be performed independently. Thus, for each local node, the covariance recursion is processed at the fusion center with the knowledge of local posterior estimate and using system/measurement dynamics. An important step for the recursion is to initialize the covariance at the FC using the knowledge of local estimate, for which, the CMKF framework is revisited. To generalize the initialization to any sensor, first-order and second-order approximation of the CMKF are subsequently derived.

Though the work-load at the FC is increased, it is negligible compared to the procedure used for association and fusion. To investigate the efficacy of the proposed method, two scenarios were simulated. One of which is a real-world tracking of 20 targets using three mono-static radars in the presence of clutter, and the other is a qualitative comparison of reconstructed and original (local) covariance. The results suggest that fusion without covariance displays insignificant difference and performs as expected. The next step in this research direction is to examine the problem

with non-observable systems such as bearings-only tracking (BOT) and covariance reconstruction in the presence of local sensor anomalies.

## Chapter 5

# Trigonometric Moments of a Generalized von Mises Distribution And Their Application in 2-D Range-Only Tracking

### 5.1 Abstract

A 2D range-only tracking scenario is non-trivial due to two main reasons. First, when the states to be estimated are in Cartesian coordinates, the uncertainty region is multi-modal. The second reason is that the probability density function of azimuth conditioned on range takes the form of a generalized von Mises distribution, which is hard to tackle. Even in the case of implementing a uni-modal Kalman filter, one needs expectations of trigonometric functions of conditional bearing density, which

are not available in the current literature. This chapter focuses on this exact problem. We prove that the trigonometric moment (circular moment) of the azimuth density conditioned on range can be computed as an infinite series, which can be sufficiently approximated by relatively few terms in summation. The solution can also be generalized to any order of the trigonometric moment.

This important result can provide an accurate depiction of the conditional azimuth density in 2D range-only tracking geometries. We also present a simple optimization problem that results in deterministic samples of conditional azimuth density from the knowledge of its circular moments, which leads to a new filtering application in such scenarios. The results are shown in a two-dimensional simulation, where the range-only observer maneuvers to make the system observable. The results prove that the method is feasible in such applications.

## 5.2 Introduction

Tracking with both azimuth and range measurements is considered a benchmark problem in target tracking scenarios. Due to the evolution of low-cost sensing and embedded technologies, tracking with range-only measurements has become a recent interest in the last decade. Also, due to advances in radio frequency technology, measuring the range between beacons and transponders is inexpensive and efficient. This has led to a tremendous use of range-only measurements in wireless sensor networks (WSN) [52]. In this work, we consider a single-sensor 2D range-only target motion analysis, which is considered impractical in long-range due to three primary reasons.

- (1) Gaussian assumption is invalid for most scenarios since the uncertainty region

resembles a concave shape. For a sensor with a large field-of-view (FOV) and low standard deviation, this cannot be sufficiently approximated with an ellipsoid [93, 31, 30].

- (2) The density of azimuth conditioned on range is extremely hard to simulate due to complicated expressions and non-trivial integrals.
- (3) While the observability conditions have been established for observer motion with several legs, there is no generalized solution for the observability of range-only target motion analysis [78, 88].

The first problem has been solved in various cases by either parameterizing azimuth [82] or using a multi-modal density instead of Gaussian. A Gaussian mixture approach was used in [31] and [30] wherein both the prior density and the conditional azimuth density were assumed to be Gaussian mixtures. The authors used the EM algorithm for Gaussian mixture approximation based on the knowledge of azimuth density, but the azimuth density itself was sampled uniformly.

Due to observability issues, range-only tracking is seldom used except in WSNs, where measurements from multiple nodes are fused together to achieve observability. As mentioned before, unlike bearing-only target motion analysis, observability in range-only scenarios does not possess a general solution for observer motion. Pillion et al. in [78] analyzed leg-by-leg observer trajectory and found that a multi-leg scenario might be observable if the range-rate is constant. The analysis was performed after assuming a nearly constant velocity (NCV) model for the target.

In this chapter, we focus on the first two problems, which are based on the multi-modal shape of the density of azimuth conditioned on range measurements in the two-dimensional case. Specifically, we derive a generalized solution for the conditional

integral of the form,

$$\mathbb{E}[\cos(m\theta)|r] = \int_0^{2\pi} \cos(m\theta)p(\theta|r)d\theta, \quad m = 1, 2, \dots \quad (5.2.1a)$$

$$\mathbb{E}[\sin(m\theta)|r] = \int_0^{2\pi} \sin(m\theta)p(\theta|r)d\theta, \quad m = 1, 2, \dots \quad (5.2.1b)$$

Such integrals are better known as trigonometric moments or circular moments in literature. Here,  $\theta$  is the azimuth,  $r$  is the range,  $m$  is a positive integer, and  $p(\cdot)$  represents the probability density function. We show that the conditional density  $p(\theta|r)$  resembles a generalized form of the von Mises distribution [39]. As per the author's knowledge, trigonometric moments of such a distribution do not exist in the current literature and hence are a powerful tool to infer the shape of conditional azimuth density in a range-only tracking scenario.

The rest of the chapter is organized as follows : In Section 5.3 we provide a general overview of the range-only tracking problem along with a modified measurement model. Generalized update equations for range-only tracking are presented in Section 5.4. In this section, we present an unimodal Kalman filter-type recursion for range-only scenarios. The true range density and the conditional azimuth densities are derived in Section 5.5. We solve the integrals in eqn. (5.2.1) in Section 5.6 and subsequently apply it to find deterministic samples of the conditional density in Section 5.7. The derivations are tested against numerical quadrature and a 2D simulation in Section 5.8. Finally, the chapter is concluded in Section 5.9.

### 5.3 Problem Formulation

Consider a linear Gaussian system that can be represented in a discrete-time state-space formulation as,

$$\mathbf{x}_k^r = \mathbf{F}_k \mathbf{x}_{k-1}^r + \mathbf{F}_k \mathbf{x}_{k-1}^o - \mathbf{x}_k^o + \mathbf{w}_k \quad (5.3.1)$$

Where  $\mathbf{x}_k^r$  is the target state vector relative to the observer state  $\mathbf{x}_k^o$  at time  $k$ .  $\mathbf{F}_k$  is the state transition matrix, and  $\mathbf{w}_k$  is the process noise involved. We assume that the process noise is zero-mean Gaussian  $\sim \mathcal{N}(\mathbf{w}_k; \mathbf{0}, \mathbf{Q}_k)$ , with  $\mathbf{Q}_k$  as the process noise covariance matrix. We also assume the whiteness of the process noise, which also imparts Markovian properties to the system. Hence,

$$\mathbb{E}[\mathbf{w}_l \mathbf{w}_m] = \delta_{lm} \mathbf{Q}_l \quad (5.3.2)$$

where  $\delta_{lm}$  is the Kronecker delta. The measurement in our case is range-only which is often governed by the additive noise model in literature as [13, 17],

$$r_k = \|\mathbf{H} \mathbf{x}_k^r\| + v_k \quad (5.3.3)$$

Where  $\mathbf{H}$  is an appropriate matrix which captures the positional components from  $\mathbf{x}_k$ , and  $v_k$  is the adjoining measurement noise,  $\sim \mathcal{N}(v_k, 0; \sigma_r^2)$ , such that it is white and uncorrelated with  $\mathbf{w}_k$ . However, we use a slightly modified version of this model since it is convenient for our analysis (discussed in the next subsection).

The objective is to estimate the state-space vector  $\mathbf{x}_k$  at each time  $k$  such that the solution is optimal in some sense. The range measurement model we choose is

based on adding noise before norm calculation, which is also asserted in [31], details of which are discussed next.

### 5.3.1 Range Measurement Model

Since range can never be negative, additive Gaussian noise makes no sense when working in short-range scenarios. For e.g., if the true range is say,  $r_k^t$  meters and the standard deviation of the measurement noise is say,  $\sigma_r$  meters, then the probability that the resulting range measurement is negative (according to eqn (5.3.3)) is,

$$\begin{aligned} \Pr(r_k < 0) &= \int_{-\infty}^0 \mathcal{N}(r; r_k^t, \sigma_r) dr = \phi\left(\frac{-r_k^t}{\sigma_r}\right) \\ &= \frac{1}{2} \left[ 1 - \operatorname{erf}\left(\frac{r_k^t}{\sigma_r \sqrt{2}}\right) \right]. \end{aligned} \quad (5.3.4)$$

Where  $r_k^t = \|\mathbf{H}\mathbf{x}_k\|$  is the true range,  $\phi(\cdot)$  is the cumulative distribution function (CDF) of a standard normal variable, and  $\operatorname{erf}(\cdot)$  denotes the error function. Thus, the probability increases with decreasing range and is never theoretically zero. Therefore, we advocate the use of additive Gaussian term inside the norm operator,

$$r_k = \|\mathbf{H}\mathbf{x}_k + \mathbf{v}_k\| \quad (5.3.5)$$

where  $\mathbf{v}_k \sim \mathcal{N}(0, \sigma_r^2 \mathbf{I})$  with  $\mathbf{I}$  being the identity matrix of appropriate dimension. This model however, gives rise to the following Ricean density for conditional range [81],

$$p_r(r_k | \mathbf{x}_k) = \frac{r_k}{\sigma_r^2} \exp\left(-\frac{(r_k)^2 + \|\mathbf{H}\mathbf{x}_k\|^2}{2\sigma_r^2}\right) I_0\left(\frac{r_k \|\mathbf{H}\mathbf{x}_k\|}{\sigma_r^2}\right) \quad (5.3.6)$$



Where  $I_0(\cdot)$  is the modified Bessel function of order 0. In contrast, the Gaussian density of the original measurement, conditioned on the truth is,

$$p_g(r_k|\mathbf{x}_k) = \frac{1}{\sigma_r\sqrt{2\pi}} \exp\left(-\frac{(r_k - \|\mathbf{H}\mathbf{x}_k\|)^2}{2\sigma_r^2}\right) \quad (5.3.7)$$

It can be observed that asymptotically, as  $\frac{r_k}{\sigma_r}$  increases, the difference  $p_r(\cdot) - p_g(\cdot)$  converges to,

$$p_r(r_k|\mathbf{x}_k) - p_g(r_k|\mathbf{x}_k) \longrightarrow p_g(r_k|\mathbf{x}_k) \left[ \sqrt{\frac{r_k}{r_k^t}} - 1 \right] \quad (5.3.8)$$

Therefore, practically, at large ranges and moderate standard deviations,  $r_k \approx r_k^t$ , and the densities are indistinguishable. The reason for choosing such a range model is only analytic convenience in arriving at closed-form solutions. To see this, consider linear measurements of position with Gaussian additive noise,

$$\mathbf{y}_k = r_k^t \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} + \mathbf{y}_k^v \quad (5.3.9)$$

Where  $\theta$  is the azimuth measured counter-clockwise from the x-axis. It is easier to observe that the density of range measurement,  $r_k = \|\mathbf{y}_k\|$  will coincide with that in Eqn. (5.3.6).

## 5.4 Update Equations for Range-Only Tracking

The posterior density of the state  $\mathbf{x}_k$  conditioned on range  $r_k$  can be represented as the marginalization of azimuth  $\theta$  as

$$p(\mathbf{x}_k|r_k) = \int_0^{2\pi} p(\mathbf{x}_k, \theta_k|r_k) d\theta_k = \int_0^{2\pi} p(\mathbf{x}_k|\theta_k, r_k) p(\theta_k|r_k) d\theta_k \quad (5.4.1)$$

The interpretation of the above equations is as follows. Assume that a linear measurement is available as in eqn. (5.3.9). Then, the linear Kalman update states,

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k [\mathbf{y}_k - \hat{\mathbf{y}}_k] = (\mathbf{I} - \mathbf{K}\mathbf{H}) \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k r_k \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} \quad (5.4.2)$$

where  $\mathbf{y}_{k|k-1}$  is the predicted measurement which can be calculated using the prior density. Since the azimuth information is not available, we find the posterior estimate by conditioning on range.

$$\hat{\mathbf{x}}_{k|k} = \mathbb{E} [\hat{\mathbf{x}}_{k|k}|r_k] = (\mathbf{I} - \mathbf{K}\mathbf{H}) \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k r_k \mathbb{E} [\mathbf{b}_k|r_k] \quad (5.4.3)$$

Where,  $\mathbf{b}_k = \begin{bmatrix} \cos(\theta_k) & \sin(\theta_k) \end{bmatrix}^T$ . Similarly, the corresponding covariance can be calculated as,

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}\mathbf{H}) \mathbf{P}_{k|k-1} + r_k^2 \mathbf{K}_k \mathbf{P}_{b,k} \mathbf{K}_k^T \quad (5.4.4)$$

where, the associated covariance due to azimuth uncertainty,  $\mathbf{P}_{b,k}$  is quantified as,

$$\mathbf{P}_{b,k} = \mathbb{E} \left[ (\mathbf{b}_k - \mathbb{E}[\mathbf{b}_k]) (\mathbf{b}_k - \mathbb{E}[\mathbf{b}_k])^T | r_k \right] \quad (5.4.5)$$

These equations follow as a direct consequence of eqn. (5.4.1). Though the Kalman filter update equations look promising for generating a near-optimal estimate, there is, however, a missing piece of information. The conditional expectations in the above equation assert that the azimuth density (conditioned on range) is considered a symmetric uni-modal distribution, which is unfortunately not true in all scenarios. The famous contact lens problem in a range-only scenario generates a conic-shaped distribution for the state estimate uncertainty, which has to be dealt with accordingly. The issue has been solved by splitting the uni-modal density into a Gaussian approximation, which can be recursively predicted and updated to generate a multi-modal posterior density.

In this work, we focus on deriving the exact expression for the conditional expectation in eqn. (5.4.3) and eqn. (5.4.5) and its generalization. The exact expressions will prove to be helpful in generating accurate multi-modal approximations to the non-trivial conditional density,  $p(\theta_k | r_k)$ , as we see in the subsequent sections.

## 5.5 Conditional Azimuth Density

The conditional azimuth density can be represented as,

$$p(\theta_k | r_k) = \frac{p(\theta_k, r_k)}{\int_0^{2\pi} p(\theta_k, r_k) d\theta_k} \quad (5.5.1)$$

Assuming that the true position at time  $k$  is Gaussian distributed with predicted estimate as the mean,  $\mathbf{y}_k \sim \mathcal{N}(\mathbf{y}_k; \hat{\mathbf{y}}_k, \mathbf{V}_k)$  as in eqn. (5.3.9), and using transformation of variables, the joint density can be transformed from Cartesian to polar coordinates,

$$p(\theta_k, r_k) = \frac{r_k}{\sqrt{|2\pi\mathbf{V}_k|}} \exp \left[ -\frac{1}{2} (r_k^2 \mathbf{b}_k^T \mathbf{V}_k^{-1} \mathbf{b}_k + \hat{\mathbf{y}}_k^T \mathbf{V}_k^{-1} \hat{\mathbf{y}}_k) \right] \exp \left[ -\frac{1}{2} (-2r_k \mathbf{b}_k^T \mathbf{V}_k^{-1} \hat{\mathbf{y}}_k) \right] \quad (5.5.2)$$

where  $\mathbf{V}_k = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R}_k$  is the predicted measurement covariance. The joint density can be rewritten as,

$$p(\theta_k, r_k) = \frac{r_k}{\sqrt{|2\pi\mathbf{V}_k|}} \exp \left[ -\frac{1}{2} \left( r_k^2 \left( \frac{a}{2} + \frac{c}{2} \right) + \hat{\mathbf{y}}_k^T \mathbf{V}_k^{-1} \hat{\mathbf{y}}_k \right) \right] \times \exp \left[ r_k \|p, q\| \cos(\theta_k - \phi_1) + A_3 r_k^2 \cos(2\theta_k + \phi_2) \right] \quad (5.5.3)$$

where following notations are used,

$$\begin{bmatrix} a & b \\ b & c \end{bmatrix} = \mathbf{V}_k^{-1}, \quad \begin{bmatrix} p \\ q \end{bmatrix} = \mathbf{V}_k^{-1} \hat{\mathbf{y}}_k, \quad \phi_1 = \tan^{-1} \left( \frac{q}{p} \right) \\ A_3 = \left\| \left( \frac{c}{4} - \frac{a}{4} \right), \frac{b}{2} \right\|, \quad \phi_2 = \tan^{-1} \left( \frac{b/2}{\left( \frac{c}{4} - \frac{a}{4} \right)} \right) \quad (5.5.4)$$

Note that the notation  $\|\alpha, \beta\|$  means the norm of a vector containing  $\alpha$  and  $\beta$ .

### 5.5.1 True Range Density

The true range density can be computed by solving the integral

$$\begin{aligned} p(r_k) &= \int_0^{2\pi} p(\theta_k, r_k) d\theta_k \\ &= \kappa(r_k) \int_0^{2\pi} \exp \left[ r_k \|p, q\| \cos(\theta_k - \phi_1) + A_3 r_k^2 \cos(2\theta_k + \phi_2) \right] d\theta_k, \end{aligned} \quad (5.5.5)$$

where  $\kappa(r_k)$  contain all the terms independent of  $\theta_k$ . The integral has been solved by Weil for the diagonal case in [103]. For the general case of eqn. (5.5.5), the integral can be solved as a series expansion (see proof in appendix) wherein the density can be calculated by a relatively fewer number of terms. The resulting range density which is also the denominator in eqn. (5.5.1) is,

$$\begin{aligned} p(r_k) &= \kappa(r_k) 2\pi \left[ I_0(A_3 r_k^2) I_0 \left( r_k \sqrt{A_1^2 + A_2^2} \right) + \right. \\ &\quad \left. 2 \sum_{k=1}^{\infty} I_k(A_3 r_k^2) I_{2k} \left( r_k \sqrt{A_1^2 + A_2^2} \right) \cos(2k\psi) \right] \end{aligned} \quad (5.5.6)$$

where  $I_v(\cdot)$  is the modified Bessel function of the first kind of order  $v$ , and,

$$A_1 = \|p, q\| \cos\left(\frac{\phi_2}{2} + \phi_1\right), \quad A_2 = \|p, q\| \sin\left(\frac{\phi_2}{2} + \phi_1\right), \quad \psi = \tan^{-1}\left(\frac{A_2}{A_1}\right) \quad (5.5.7)$$

The resulting conditional azimuth density is then,

$$p(\theta_k | r_k) = \frac{\kappa(r_k)}{p(r_k)} \times \exp \left[ r_k \|p, q\| \cos(\theta_k - \phi_1) + A_3 r_k^2 \cos(2\theta_k + \phi_2) \right] \quad (5.5.8)$$

where  $p(r_k)$  is given in eqn. (5.5.6), note that  $\kappa(r_k)$  will be canceled out. The remaining denominator in eqn. (5.5.8) is

$$\text{den} = 2\pi \left[ I_0(A_3 r_k^2) I_0\left(r_k \sqrt{A_1^2 + A_2^2}\right) + 2 \sum_{k=1}^{\infty} I_k(A_3 r_k^2) I_{2k}\left(r_k \sqrt{A_1^2 + A_2^2}\right) \cos(2k\psi) \right], \quad (5.5.9)$$

which can be approximated by a fewer number of terms in summation.

## 5.6 Expectations of Conditional Azimuth Density

To calculate the optimal solution in eqns. (5.4.3) and (5.4.5), it is necessary to calculate the expectations of the form  $\mathbb{E}[\cos(\theta_k)|r_k]$ ,  $\mathbb{E}[\sin(\theta_k)|r_k]$ , and others. We show that it is possible to compute the generalized expectations of the form  $\mathbb{E}[\cos(m\theta_k)|r_k]$  and  $\mathbb{E}[\sin(m\theta_k)|r_k]$  for any positive integer  $m$ .

$$\begin{aligned} \mathbb{E}[\cos(m\theta_k)|r_k] &= \int_0^{2\pi} \cos(m\theta_k) p(\theta_k|r_k) d\theta_k = I_{\text{gen, cos}}(m) \\ &\propto \int_0^{2\pi} \cos(m\theta_k) \exp\left[r_k \|p, q\| \cos(\theta_k - \phi_1) + A_3 r_k^2 \cos(2\theta_k + \phi_2)\right] d\theta_k \end{aligned} \quad (5.6.1)$$

where the proportionality constant is  $\text{den}^{-1}$ . In order to proceed with the integral, let's solve a simplified version of the form,

$$I_{\text{diag, cos}}(m) = \int_0^{2\pi} \cos(m\theta_k) \exp\left[r_k \|p, q\| \cos(\theta_k - \phi_1) + A_3 r_k^2 \cos(2\theta_k)\right] d\theta_k \quad (5.6.2)$$

and,

$$I_{\text{diag},\sin}(m) = \int_0^{2\pi} \sin(m\theta_k) \exp \left[ r_k \|p, q\| \cos(\theta_k - \phi_1) + A_3 r_k^2 \cos(2\theta_k) \right] d\theta_k \quad (5.6.3)$$

Note that the difference is in the inclusion of  $\phi_2$ . We use a subscript ‘diag’ for these integrals since this is the case for diagonal covariance matrix  $\mathbf{V}_k$ . It is proven in the appendix that these integrals result in,

$$I_{\text{diag},\cos}(m) = 2\pi \sum_{j=-\infty}^{\infty} \frac{1}{2} I_j(A_3 r_k^2) \left[ I_{2j+m}(Dr_k) \cos((2j+m)\psi) + I_{2j-m}(Dr_k) \cos((2j-m)\psi) \right] \quad (5.6.4)$$

and,

$$I_{\text{diag},\sin}(m) = 2\pi \sum_{j=-\infty}^{\infty} \frac{1}{2} I_j(A_3 r_k^2) \left[ I_{2j+m}(Dr_k) \sin((2j+m)\psi) - I_{2j-m}(Dr_k) \sin((2j-m)\psi) \right] \quad (5.6.5)$$

where  $D = \sqrt{A_1^2 + A_2^2}$  and  $\tan(\psi) = \frac{A_2}{A_1}$ .

Moving forward, note that the term inside the summation is an even function of summation index  $j$ . Therefore, the infinite series can be reduced to positive indices only.

$$I_{\text{diag},\cos}(m) = 2\pi \left[ I_0(A_3 r_k^2) I_m(Dr_k) \cos(m) + \sum_{j=1}^{\infty} I_j(A_3 r_k^2) \times \right. \\ \left. (I_{2j+m}(Dr_k) \cos((2j+m)\psi) + I_{2j-m}(Dr_k) \cos((2j-m)\psi)) \right], \quad (5.6.6)$$

$$I_{\text{diag},\sin}(m) = 2\pi \left[ I_0(A_3 r_k^2) I_m(Dr_k) \sin(m) + \sum_{j=1}^{\infty} I_j(A_3 r_k^2) \times \right. \\ \left. (I_{2j+m}(Dr_k) \sin((2j+m)\psi) - I_{2j-m}(Dr_k) \sin((2j-m)\psi)) \right], \quad (5.6.7)$$

which reduces the computation time by half,

The intergal for the general case in eqn. (5.6.1) can be converted to that in eqns. (5.6.2) and (5.6.3) using the substitution  $2\theta_k + \phi_2 = 2u$ . Which results in the following expressions for the integrals of our interest,

$$I_{\text{gen},\cos}(m) = \mathbb{E}[\cos(\theta_k)]_{p(\theta_k|r_k)} = \frac{2\pi}{\text{den}} \left[ \cos\left(m \frac{\phi_2}{2}\right) I_{\text{diag},\cos} + \sin\left(m \frac{\phi_2}{2}\right) I_{\text{diag},\sin} \right] \quad (5.6.8)$$

$$I_{\text{gen},\sin}(m) = \mathbb{E}[\sin(\theta_k)]_{p(\theta_k|r_k)} = \frac{2\pi}{\text{den}} \left[ \cos\left(m \frac{\phi_2}{2}\right) I_{\text{diag},\sin} - \sin\left(m \frac{\phi_2}{2}\right) I_{\text{diag},\cos} \right] \quad (5.6.9)$$

Where  $I_{\text{diag},\cos}$  and  $I_{\text{diag},\sin}$  are given in eqns. (5.6.6) and (5.6.7) respectively. Since  $r_k$  is generally of the order of thousands of meters, attempting to calculate the Bessel functions in above expressions directly might result in floating point errors. One way is to use the following approximation for the ratio of Bessel functions with the same argument [76, 104, 77],

$$\frac{I_N(x)}{I_0(x)} \approx \exp\left(-\frac{N^2}{2x}\right), \quad x \gg N^2 \gg 1 \quad (5.6.10)$$

The approximation tends to get accurate with increasing argument  $x$ . Under this assumption, the error was found to be of the order  $\mathcal{O}\left(\frac{N^4}{x^2}\right)$  [96].

Dividing the numerator and denominator in eqns. (5.6.8) and (5.6.9) by  $I_0(A_3 r_k^2) I_0(r_k D)$ ,



where  $D = \sqrt{A_1^2 + A_2^2}$ , and using eqn. (5.6.10), the result is,

$$I_{\text{gen},\cos}(m) = \frac{\cos\left(m\frac{\phi_2}{2}\right) I_{\text{diag},\cos}^n(m) + \sin\left(m\frac{\phi_2}{2}\right) I_{\text{diag},\sin}^n(m)}{1 + 2 \sum_{k=1}^{\infty} \exp\left(-\frac{k^2}{2A_3 r_k^2}\right) \exp\left(-\frac{(2k)^2}{Dr_k}\right) \cos(2k\psi)} \quad (5.6.11)$$

$$I_{\text{gen},\sin}(m) = \frac{\cos\left(m\frac{\phi_2}{2}\right) I_{\text{diag},\sin}^n(m) - \sin\left(m\frac{\phi_2}{2}\right) I_{\text{diag},\cos}^n(m)}{1 + 2 \sum_{k=1}^{\infty} \exp\left(-\frac{k^2}{2A_3 r_k^2}\right) \exp\left(-\frac{(2k)^2}{Dr_k}\right) \cos(2k\psi)} \quad (5.6.12)$$

where  $I_{\text{diag},\cos}^n(I_{\text{diag},\sin}^n)$  is same as  $I_{\text{diag},\cos}(I_{\text{diag},\sin})$  normalized with  $I_0(A_3 r_k^2) I_0(r_k D)$ , given by the following expression,

$$I_{\text{diag},\cos}^n(m) = \frac{I_{\text{diag},\cos}(m)}{I_0(A_3 r_k^2) I_0(r_k D)} = \sum_{j=0}^{\infty} \exp\left(-\frac{j^2}{2A_3 r_k^2}\right) \times \left[ \exp\left(-\frac{(2j+m)^2}{2Dr_k}\right) \cos(2j+m) + \exp\left(-\frac{(2j-m)^2}{2Dr_k}\right) \cos(2j-m) \right] \quad (5.6.13)$$

$$I_{\text{diag},\sin}^n(m) = \frac{I_{\text{diag},\sin}(m)}{I_0(A_3 r_k^2) I_0(r_k D)} = \sum_{j=0}^{\infty} \exp\left(-\frac{j^2}{2A_3 r_k^2}\right) \times \left[ \exp\left(-\frac{(2j+m)^2}{2Dr_k}\right) \sin(2j+m) - \exp\left(-\frac{(2j-m)^2}{2Dr_k}\right) \sin(2j-m) \right] \quad (5.6.14)$$

Note that the condition  $x \gg N^2 \gg 1$  is crucial for the approximation to be meaningful. Other ways for computing Bessel function ratios are mentioned in [7] or, directly by using a high precision computing mechanism like Matlab's<sup>©</sup> `vpa`.

### 5.6.1 Trigonometric Moments

Once the generalized expectations are formulated, the covariance and higher-order moments can be calculated using known trigonometric identities. For instances, the

mean is given by,

$$\mathbb{E}[\mathbf{b}_k | r_k] = \mathbb{E} \left[ \begin{array}{c|c} \cos(\theta_k) & \\ \sin(\theta_k) & \end{array} \middle| r_k \right] = \begin{bmatrix} \mathbf{I}_{\text{gen,cos}}(1) \\ \mathbf{I}_{\text{gen,sin}}(1) \end{bmatrix} \quad (5.6.15)$$

The covariance  $\mathbf{P}_{b,k}$  in eqn. (5.4.5) is given by the formula,

$$\mathbf{P}_{b,k} = \mathbb{E} \left[ \begin{array}{cc|c} \cos^2(\theta_k) & \cos(\theta) \sin(\theta_k) & \\ \sin(\theta_k) \cos(\theta_k) & \sin^2(\theta_k) & \end{array} \middle| r_k \right] - \mathbb{E}[\mathbf{b}_k] \mathbb{E}[\mathbf{b}_k]^T \quad (5.6.16)$$

where the expectations can be calculated element-wise.

$$\mathbb{E} [\cos^2(\theta_k) | r_k] = \frac{1}{2} + \frac{\mathbf{I}_{\text{gen,cos}}(2)}{2}, \quad (5.6.17)$$

$$\mathbb{E} [\sin^2(\theta_k) | r_k] = \frac{1}{2} - \frac{\mathbf{I}_{\text{gen,cos}}(2)}{2} \quad (5.6.18)$$

$$\mathbb{E} [\cos(\theta_k) \sin(\theta_k) | r_k] = \frac{\mathbf{I}_{\text{gen,sin}}(2)}{2} \quad (5.6.19)$$

It is even possible to calculate higher order powered moments using power reduction trigonometric identities. For instance,

$$\mathbb{E} [\cos^3(\theta_k) | r_k] = \frac{3\mathbf{I}_{\text{gen,cos}}(1) + \mathbf{I}_{\text{gen,cos}}(3)}{4} \quad (5.6.20)$$

$$\mathbb{E} [\cos^4(\theta_k) | r_k] = \frac{3 + 4\mathbf{I}_{\text{gen,cos}}(2) + \mathbf{I}_{\text{gen,cos}}(4)}{8} \quad (5.6.21)$$

$$\mathbb{E} [\sin^3(\theta_k) | r_k] = \frac{3\mathbf{I}_{\text{gen,sin}}(1) - \mathbf{I}_{\text{gen,sin}}(3)}{4} \quad (5.6.22)$$

$$\mathbb{E} [\sin^4(\theta_k) | r_k] = \frac{3 - 4\mathbf{I}_{\text{gen,cos}}(2) + \mathbf{I}_{\text{gen,cos}}(4)}{8} \quad (5.6.23)$$

Thus, once the recursive functions,  $I_{\text{gen,cos}}(m)$  and  $I_{\text{gen,sin}}(m)$  are constructed, it is trivial to calculate any higher order moment.

## 5.7 Deterministic Sampling of Conditional Azimuth Density and Application to Tracking

In the previous section, we proved that the azimuth density can be computed in closed form, and their trigonometric moments (also read as circular moments) exist as infinite series that can be approximated well using a very small number of terms (see simulation). In practical range-filtering, the Gaussian approximation is seldom useful due to the contact-lens problem, and thus, it is essential to capture the multimodality of conditional azimuth density.

A viable solution to this difficulty is to approximate the conditional azimuth density using its deterministic samples and then proceed with eqn. (5.4.1) to get the posterior estimate of the target state. Deterministic samples from circular moments have been previously proposed in [58, 45, 31] but we take a slightly straightforward optimization-based approach here, which is explained as follows.

The wrapped Dirac distribution is a perfect candidate for the deterministic sampling procedure. The distribution is given by,

$$p_d(\theta) = \sum_{l=1}^L \gamma_l \delta(\theta - \hat{\theta}_l) \quad (5.7.1)$$

where  $\hat{\theta}_l \in [0, 2\pi]$  are the sigma-points position and  $\gamma_l \in [0, 1]$  are the sigma-points weights corresponding to  $\theta_l$  such that  $\sum_l \gamma_l = 1$ . Thus, there are  $2L$  variable to be

calculated for Dirac distribution consisting of  $L$  components. The  $m^{\text{th}}$  trigonometric moments of a wrapped Dirac distribution are,

$$\mathbb{E}[\cos(m\theta)]_{p_d(\theta)} = \sum_{l=1}^L \gamma_l \cos(m\hat{\theta}_l), \quad (5.7.2)$$

$$\mathbb{E}[\sin(m\theta)]_{p_d(\theta)} = \sum_{l=1}^L \gamma_l \sin(m\hat{\theta}_l) \quad (5.7.3)$$

For approximation of  $p(\theta_k|r_k)$ , we substitute the L.H.S in the above equations with our series-approximated  $m^{\text{th}}$  order moments and solve for  $\gamma = [\gamma_1, \gamma_2, \dots, \gamma_L]$  and  $\theta = [\theta_1, \theta_2, \dots, \theta_L]$ . Thus, a total of  $2M$  equations need to be solved for an  $L$ -component distribution, where  $M$  is the highest order considered for computation. Mathematically, we solved the following norm-minimization problem to arrive at optimal values of  $\gamma$  and  $\theta$ .

$$\gamma^*, \theta^* = \arg \min_{\gamma, \theta} \left\| \begin{array}{c} \vdots \\ I_{\text{gen}, \cos}(m) - \sum_{l=1}^L \gamma_l \cos(m\hat{\theta}_l) \\ I_{\text{gen}, \sin}(m) - \sum_{l=1}^L \gamma_l \sin(m\hat{\theta}_l) \\ \vdots \end{array} \right\|, \quad \forall m = 1, 2, \dots, M \quad (5.7.4)$$

such that,

$$1 \geq \gamma_l \geq 0; \quad 2\pi \geq \theta_l \geq 0 \quad \forall l = 1, 2, \dots, L$$

The optimal values of  $\gamma$  obtained from above are then normalized such that  $\sum_l \gamma_l^* = 1$ .

The resulting azimuth density and corresponding values of  $\gamma, \hat{\theta}$  for following values of  $\mathbf{V}_k$  and  $\hat{\mathbf{y}}_k$  is plotted in Fig. 5.1.

$$\mathbf{V}_k = q \times 10^3 \begin{bmatrix} 7 & 2 \\ 2 & 1 \end{bmatrix}, \quad \hat{\mathbf{y}}_k = \begin{bmatrix} -50 \\ 20 \end{bmatrix} \quad (5.7.5)$$

Where  $q$  is any positive real number. For comparison, the symmetric sampling technique, which uses only the first and second order moments [58], is also plotted in Fig. 5.1. The optimization was based on 14 components in the Dirac distribution using the first 7 moments (7 moments for sin & cos respectively, thus equating 14 components). It can be seen that the optimization based approach faithfully captures the multi-modality and skewness of the original distribution. More components can be added for accuracy at the cost of computational time.

Matlab's<sup>©</sup> `fmincon` was used to minimize eqn. (5.7.4) using the sequential quadratic programming (SQP) algorithm [75]. The algorithm found an optima much faster when the system was overdetermined, which means that the number of moments (M) employed was greater than the number of component parameters (L).

### 5.7.1 Proposed Algorithm for 2D Range-Only Tracking

Assuming that a Gaussian prior  $\hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}$  is available, the proposed algorithm is as follows :

- Choose an integer  $N$  for the number of terms used in the summation of eqn. (5.6.4) and (5.6.5).

- Choose an integer  $M$  for the order of circular moments to be computed.
- Calculate the parameters  $\mathbf{V}_k, p, q, \phi_1, A_1, A_2, A_3$  in eqns. (5.5.4) and (5.5.7).
- Compute circular moments  $-\cos(\theta_k), \sin(\theta_k); \dots; \cos(M\theta), \sin(M\theta_k)$  from eqns. (5.6.8) and (5.6.9).
- From circular moments, calculate  $L$  deterministic points  $(\hat{\theta}_{l,k})$  and corresponding weights  $\gamma_l$  of the conditional azimuth density by solving the optimization problem in eqn. (5.7.4).
- Substitute the Dirac approximation in eqn. (5.4.1),

$$\begin{aligned}
p(\mathbf{x}_k|r_k) &= \int_0^{2\pi} p(\mathbf{x}_k|\theta_k, r_k)p(\theta_k|r_k)d\theta_k \\
&= \int_0^{2\pi} p(\mathbf{x}_k|\theta_k, r_k) \sum_{l=1}^L \gamma_l \delta(\theta_k - \hat{\theta}_{l,k}) d\theta_k \\
&= \sum_{l=1}^L \gamma_l \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^l, \mathbf{P}_{k|k}^l)
\end{aligned} \tag{5.7.6}$$

where the local components  $\hat{\mathbf{x}}_{k|k}^l, \mathbf{P}_{k|k}^l$  are given by,

$$\hat{\mathbf{x}}_{k|k}^l = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \hat{\mathbf{x}}_{k|k-1} + r_k \mathbf{K}_k \begin{bmatrix} \cos(\hat{\theta}_{l,k}) \\ \sin(\hat{\theta}_{l,k}) \end{bmatrix} \tag{5.7.7}$$

$$\mathbf{P}_{k|k}^l = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_{k|k-1} + r_k^2 \mathbf{K}_k \mathbf{P}_{b,k} \mathbf{K}_k^T \tag{5.7.8}$$

where the covariance  $\mathbf{P}_{b,k}$  is given in eqn. (5.6.16).

- As expected, the resulting posterior in eqn. (5.7.6) takes the form of a Gaussian

mixture. Find the resulting mean and covariance, and propagate them for the next recursion.

It is obvious that the algorithm can be made more robust by using the Gaussian mixture itself as a prior instead of its Gaussian approximation. That will, however, have an impact on the computational requirements of the algorithm since mixture reduction techniques will also have to be applied. Thus, the authors intend to employ it when efficient techniques for calculating ratios of Bessel functions and optimization in eqn. (5.7.4) have been developed. As a proof-of-concept, the simulation uses a Gaussian prior for all filtering recursions.

## 5.8 Simulation Results

The validity of the proposed formulation is simulated on two cases. First, to validate the accuracy of the solved integrals with a base-line method like quadrature. The other case presents the application of proposed formulation to a real-life tracking scenario.

### 5.8.1 Comparison with Quadrature Rule

The integrals were evaluated with Matlab's<sup>©</sup> vectorized adaptive quadrature rule [83] and found that our solution converged for a few numbers of terms in summation. This simulation is based on the following values of the assumed measurement mean  $\hat{\mathbf{y}}$  and

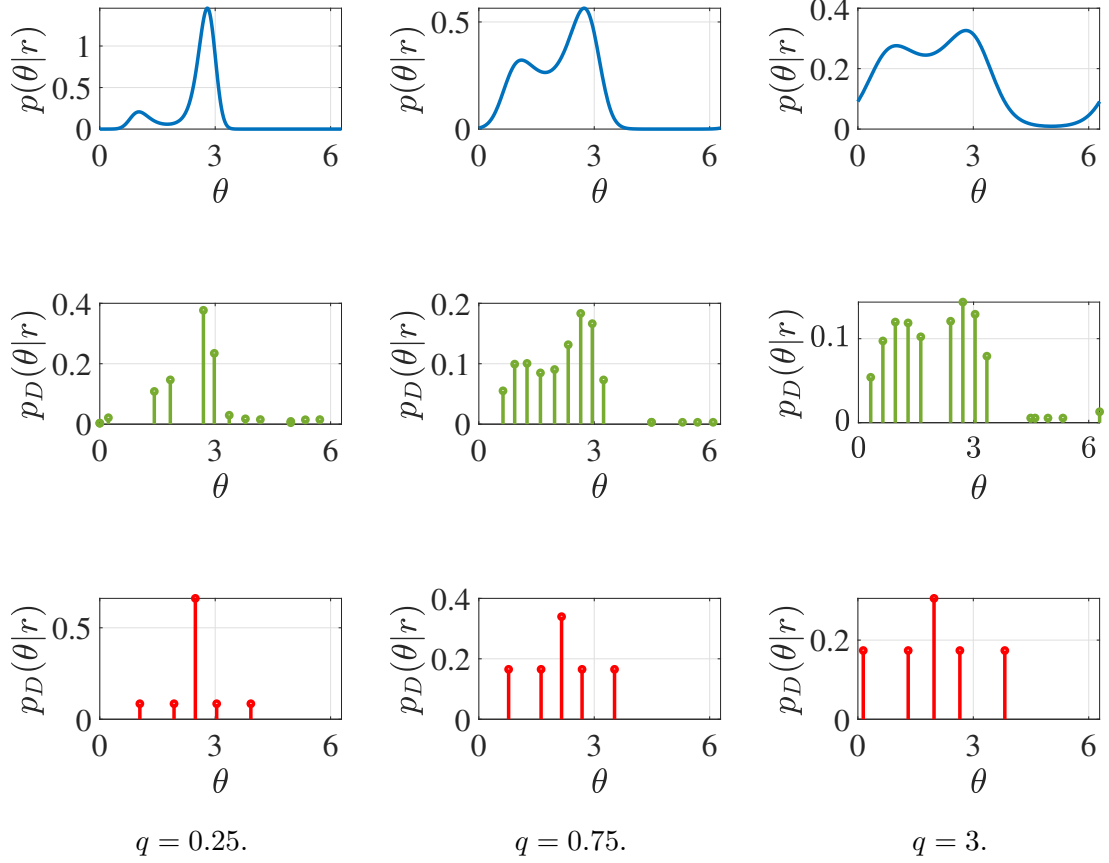


Figure 5.1: (Top) Actual Density, (Middle) Deterministic Sampling using proposed method, (Bottom) Using the technique specified in [58]; for various values of  $q$ .

measurement covariance  $\mathbf{V}$ ,

$$\hat{\mathbf{y}} = \begin{bmatrix} -11 & 20 \end{bmatrix}^T ; \quad \mathbf{V} = \begin{bmatrix} 50 & -10 \\ -10 & 50 \end{bmatrix} \quad (5.8.1)$$

Range is calculated by adding zero mean Gaussian noise with covariance  $4\mathbf{I}_2$  to  $\hat{\mathbf{y}}$ .

Integrals in eqns. (5.2.1a) and (5.2.1b) have been computed for a few values of  $m$



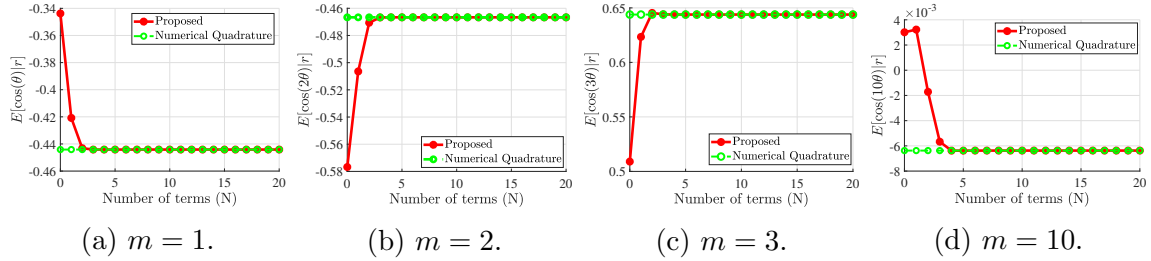


Figure 5.2: Computation of integral in eqn. (5.2.1a) for various values of moments of order  $m$ .

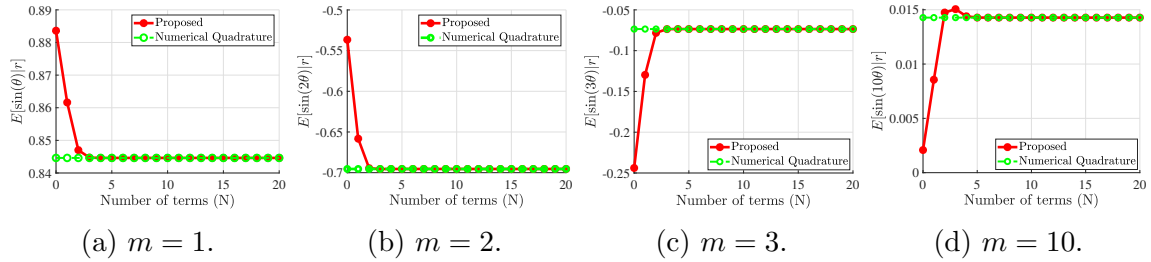


Figure 5.3: Computation of integral in eqn. (5.2.1b) for various values of  $m$ .

using the proposed formulation in eqns. (5.6.8) and (5.6.9). These values are plotted in Figs. 5.2 and 5.3 for the case cos and sin respectively against a finite number of terms  $N$  employed in the summation of eqns. (5.6.4) and (5.6.5).

The power of approximation is evident in the figures, where the proposed formulation converged with less than the first five terms of the infinite series solution. Since Matlab’s quadrature implementation is adaptive, it doesn’t take the number of steps as an argument, due to which it remains constant in Figs. 5.2 and 5.3. The corresponding error is plotted in Fig. 5.4 and also tabulated in Table 5.1. The table suggests that 10 terms are sufficient to calculate with significant accuracy. Beyond this, the error is incomprehensible for Matlab’s<sup>©</sup> double precision values. The author recommends five to ten terms as being enough for most target-tracking applications.

The computation times for calculating  $\mathbb{E}[\cos(\theta)]$  are plotted in Fig. 5.5. The

Table 5.1: Error in computation ( $m = 1$ ).

	N = 0	2	5	10	15	20
$\mathbb{E}[\cos(\tilde{\theta}) r]$	-0.07	$-2.9 \times 10^{-3}$	$-1.05 \times 10^{-8}$	$7.63 \times 10^{-15}$	$-3.12 \times 10^{-17}$	$-3.12 \times 10^{-17}$
$\mathbb{E}[\sin(\tilde{\theta}) r]$	0.04	$3.1 \times 10^{-3}$	$1.74 \times 10^{-7}$	$1.46 \times 10^{-15}$	$1.66 \times 10^{-17}$	$1.66 \times 10^{-17}$

advantage of using an exact solution is evident in Fig. 5.5a, where the quadrature rule performs 10 times slower as compared to the proposed formulation. For this example, Matlab’s `double` precision was used, as the arguments were not too large. Since modified Bessel functions increase with argument, they can assume large values as  $r$  increases (not shown here). This might return `Inf` or `NaN` while coding in Octave or similar systems. We employed Matlab’s<sup>©</sup> variable-precision arithmetic (VPA), which provides the ability to apply mathematical operations to extremely large numbers without overflowing. This employs Matlab’s symbolic toolbox<sup>©</sup> and hence slows down performance (default 32-bit precision used here). It can be seen that again, the time for the quadrature rule is fairly constant as it doesn’t depend on the number of terms as an input argument, whereas the computation times in the case of an exact formulation increase almost linearly. It can be observed from the plot that the proposed method is still advantageous to use below five terms of summation, beyond which the quadrature can be used.

### 5.8.2 Single-Sensor Range-Only Target Tracking Scenario

A single-sensor range-only tracking scenario is non-trivial to construct due to challenges in observability. We chose an existing scenario, which is discussed in [78] as conditionally observable. The observability is such that it can present a ghost target, which can cause divergence in estimation, but this effect can be nullified by taking a

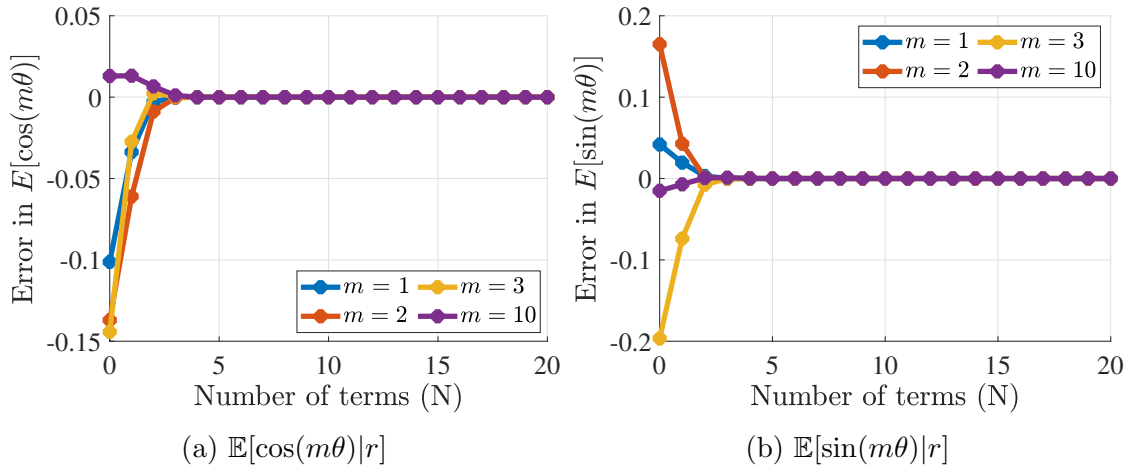


Figure 5.4: Error with respect to quadrature based computation.

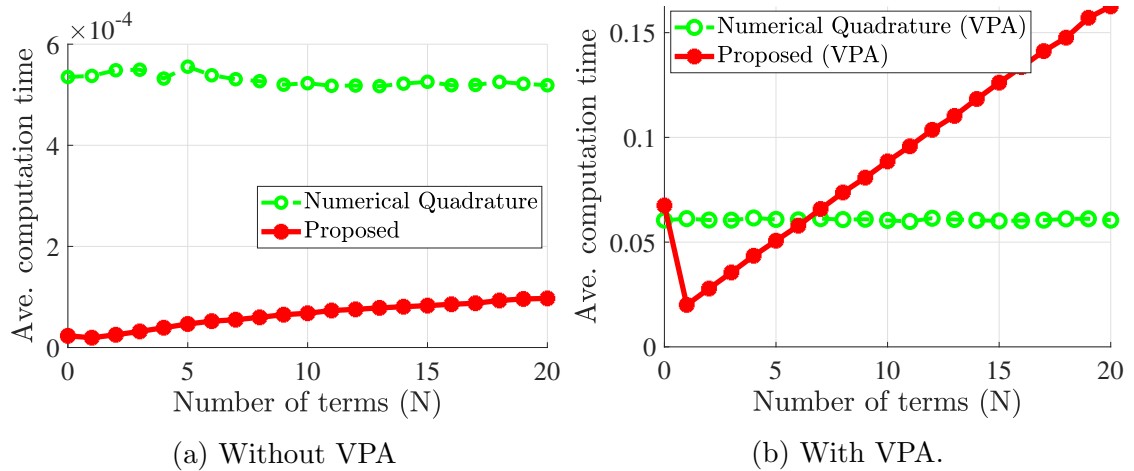


Figure 5.5: Impact of Matlab’s variable precision-arithmetic (VPA) on time-performance.

Table 5.2: Simulation Parameters

Parameters	Value
Sampling time	60 seconds.
Simulation time	30 minutes.
Initial target pos.	[7072.1, 7072.1] meters.
Target heading	225°.
Init. observer heading	170°.
Final observer heading	304°.
Target speed	15 knots.
Observer speed	5 knots.
$\tilde{q}$	$10^{-3} \text{ m}^2/\text{sec}^3$ .
$\sigma_r$	10 meters.
$\sigma_\theta$	1°.
No. of terms in summation ( $N$ )	5
Highest circular moment ( $M$ )	10
Components in Dirac approximation ( $L$ )	8

single azimuth measurement of an appropriate standard deviation at the beginning, which also constructs the prior density. Such an approach is also discussed in [82] where the RADAR begins in scan mode and subsequently switches to inverse synthetic aperture radar (ISAR) mode to retrieve high-resolution range-only measurements. In this simulation as well, we capture a single azimuth measurement to construct the prior.

The total simulation time is 30 minutes, with the rest of the parameters presented in Table 5.2. The target is assumed to be traveling with a nearly constant velocity

model injected with high process noise. Initially, it is situated 10 kilometers away from the sensor and starts moving towards it with a heading of  $225^\circ$ . The target speed is taken to be 15 knots. The observer, initially located at  $[0, 0]$  kilometers, starts moving with a heading direction of  $170^\circ$  for 15 minutes and abruptly maneuvers by changing its heading to  $304^\circ$ . Throughout the traversal, the observer's speed is fixed at 5 knots.

The relative target motion at time  $k$  is modeled using the following equation,

$$\mathbf{x}_k^r = \mathbf{F}_k \mathbf{x}_{k-1}^r + \mathbf{F}_k \mathbf{x}_{k-1}^o - \mathbf{x}_k^o + \mathbf{w}_k \quad (5.8.2)$$

where  $\mathbf{x}_k^r$  is the target state relative to the observer state  $\mathbf{x}_k^o$ .  $\mathbf{F}_k$  is the linear state transition matrix, and  $\mathbf{w}_k$  is the zero-mean process noise with covariance  $\mathbf{Q}_k$ ,

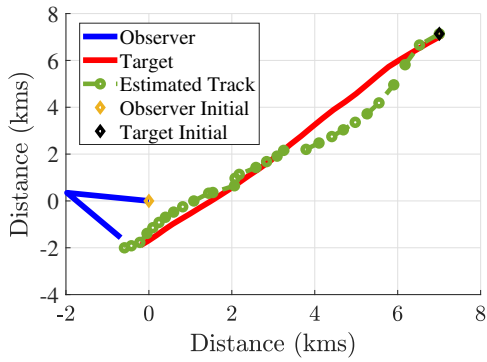
$$\mathbf{F}_k = \begin{bmatrix} \mathbf{I}_2 & T\mathbf{I}_2 \\ \mathbf{0}_2 & \mathbf{I}_2 \end{bmatrix}, \quad \mathbf{Q}_k = \tilde{q} \begin{bmatrix} \frac{T^3}{3}\mathbf{I}_2 & \frac{T^2}{2}\mathbf{I}_2 \\ \frac{T^2}{2}\mathbf{I}_2 & T\mathbf{I}_2 \end{bmatrix} \quad (5.8.3)$$

where  $\tilde{q}$  is the process noise intensity and  $T$  is the sampling-time. For measurements, the following conventional range-only measurement (noise-after-norm) is employed to check the versatility of proposed algorithm to existing systems,

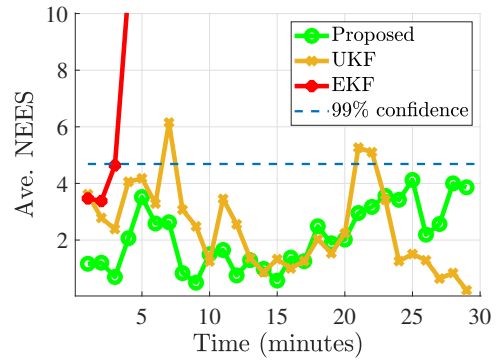
$$r_k = \sqrt{x_k^2 + y_k^2} + v_k \quad (5.8.4)$$

where  $v_k$  is a zero-mean measurement noise with covariance  $\sigma_r^2$ .

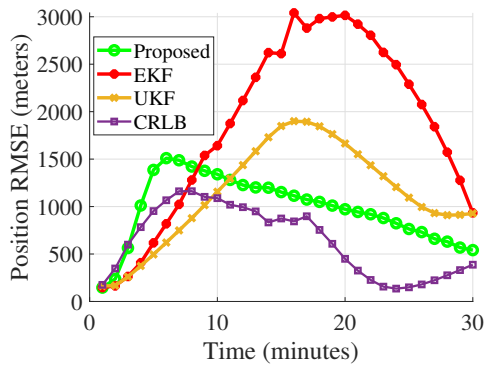
The scenario is presented in Fig. 5.6a along with the estimated track using the proposed approach. Ideally, without process noise, the observer lies along the target



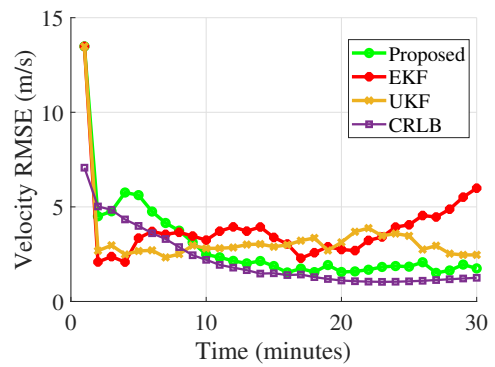
(a) True and estimated trajectories.



(b) Average NEES.



(c) Position RMSE.



(d) Velocity RMSE.

Figure 5.6: Target-Tracking Simulation Based on Proposed Approach.

path, but high process noise causes deviations in the target trajectory. As shown, the proposed approach faithfully tracks the target, with convergence achieved after the observer maneuver.

For performance evaluation, we used the root-mean-square error (RMSE) for position and velocity, as well as the normalized estimation error squared (NEES), averaged over 100 Monte-Carlo runs. The RMSE position is shown in Fig. 5.6c, where the proposed approach stands out in comparison to the extended Kalman filter (EKF) and the unscented Kalman filter (UKF) [101]. Crámer-Rao lower bound (CRLB), derived from [95] is also presented as a benchmark.

Both UKF and EKF use a linear approach by approximating the conditional distribution as a Gaussian and applying it to a linear Kalman filter. Since the standard deviation is low, the uncertainty region cannot be sufficiently approximated as an ellipsoid, due to which the minimum mean square error (MMSE) estimate is no longer captured by the mean of the Gaussian approximated posterior.

A similar trend can be seen in the velocity RMSE plot in Fig. 5.6d. The proposed algorithm stands out even when Gaussian approximation is used in place of mixture posterior density and almost reaches CRLB. Note that unlike the position RMSE plot, the CRLB does not start from the same initial point, which is due to the fact that single-point initialization was used for estimating the prior.

The last component of the simulation result is the NEES plot in Fig. 5.6b where the EKF is shown to be completely inconsistent in comparison to the UKF and the proposed approach, which are rather consistent due to the Gaussian approximation of the posterior.

## 5.9 Conclusion

In this chapter, we focused on a specific problem in the domain of 2D range-only tracking (ROT). We show that the Fourier-type integral of the azimuth conditional density exists as an infinite sum, which can produce solutions for a few numbers of summing elements. We rigorously derive the generalized density of range and then use the solution to derive the expectation of the form  $\mathbb{E}[\cos(m\theta)]$  and  $\mathbb{E}[\sin(m\theta)]$  for an arbitrary positive integer  $m$ . Using such moments, it is possible to accurately find deterministic samples of the conditional azimuth density. The formulation was compared using numerical quadrature, which proved that the proposed approach converges swiftly at a fraction of the cost.

The rest of the filtering algorithm follows from the standard total probability theorem, wherein the discrete azimuth distribution was employed, resulting in a Gaussian mixture distribution as the posterior. The results provide evidence of the benefit of using exact circular moments of the conditional density, thus, capturing the multimodality of state uncertainty.

As part of the future research, the authors would examine better approaches to calculating Bessel function ratios that, even though exist in the current literature, are almost equivalent in terms of computation time to using MATLAB's<sup>©</sup> variable precision arithmetic. Also, the optimization-based approach to calculating sigma points can be refined. Both of these approaches, when employed, would result in a much more robust and faster filtering method for 2D range-only target tracking problems.



# Chapter 6

## Conclusion

In this thesis, two of the fundamental problems in the field of track-fusion are studied. These are, namely, the fusion of non-independent marginal densities from the local nodes representing the spatial distribution of the target and the problem of fusion of local estimates with missing covariance. Both challenges find their place in real-life applications of multi-sensor target tracking. The proposed solutions have been extensively studied and applied to many real-life scenarios, proving their efficacy.

Existing track fusion strategies lack generalization. They are designed on the assumption that local track densities follow a Gaussian distribution. Existing research on multi-target target tracking suggests the use of Gaussian mixtures distribution and the Student's- $t$  process for robust filtering and extended target tracking. Attempting to fuse such tracks will result in a serious loss of information, as only the first and second order moments are captured. The proposed fusion method, harmonic mean density (HMD) fusion, provides multiple generalizations for this problem. Even though a closed-form solution is not available, a simple approximation produces near-optimal results without the requirements of memory and bookkeeping as needed in

other strategies.

Implementation using the same approximation can be applied to Gaussian mixture fusion without any changes to the framework, as in geometric mean density (GMD) fusion. The reader should be illuminated with the fact that an essential step in fusing Gaussian mixtures using GMD is to calculate the non-integer power of a Gaussian mixture, which imparts significant problems in implementation. It was shown that in both the cases of Gaussian and Gaussian mixture densities, the proposed implementation of HMD is faster than covariance intersection and its generalization, respectively.

It was also shown that for scalar estimates, the covariance intersection-like methods do not attempt any fusion, as the optimized fusion weight results in either 0 or 1, thus selecting the estimate with the least covariance. However, the HMD does not suffer from any such problems due to the presence of the spread-of-means term. Therefore, while comparing various fusion methods for implementation, a target-tracking engineer would find it advantageous over other conservative methods because of the following reasons :

- Tighter estimation error than covariance-intersection and its variants. In various simulations, the performance was found to be near-optimal in the sense of minimum-mean square error.
- No changes in the framework are required while implementing the fusion of Gaussian mixtures.
- The proposed method of optimizing fusion weights is the fastest among all conservative methods due to the requirement of only one matrix inversion. Unlike other conservative techniques, the method is also suitable for the fusion of scalar estimates.

- A sampling-based implementation ensures that any densities can be fused together as long as sampling from one of the densities is feasible. However, the calculation of fusion weights in such cases is user-dependent. Even though this is computationally expensive, it doesn't require any proposal density or adaptive tuning.

In tactical communication systems, the available bit rate between the links is sometimes not high enough such that all the sensing nodes can send full parameterization of track densities at all times. Therefore, the variance-covariance matrix is dropped before transmission. Conventionally, the best possible fusion method in such a situation is to take the average of all such estimates without knowing their covariance. Yet again, the problem of unknown cross-correlation would reflect, making the situation worse.

A novel covariance reconstruction methodology was also proposed in the thesis to tackle such issues. The idea of reconstruction was to initialize the local covariance at the fusion center using the knowledge of local posterior estimates and recursively update or predict them depending on the availability of local estimates. The converted measurement filtering was put to use here, and it was shown that a full covariance matrix can be initialized. The method was extended to both nearly-constant velocity and nearly-constant acceleration motion models.

An important result regarding the positive-definiteness of the reconstructed matrices was proven. It was shown that a similar result can be deduced using first- and second-order linearization, with some obvious loss of information during initialization. This would make the implementation feasible for a range of sensor models. Results prove that the algorithm converges in the case of the Frobenius norm and

mean square deviation metrics. An implementation in a multi-sensor, multi-target tracking scenario proves the effectiveness of the algorithm.

Chapter 4 was slightly off track from the rest of the thesis. The core research idea started with the tracking of heterogeneous tracks and developed into a novel filtering application to 2D range-only tracking. Nevertheless, the formulation presented contains a fundamental result that can be incorporated into existing systems. We proved that the conditional azimuth density in 2D range-only tracking follows a generalized von Mises distribution and calculated its trigonometric moments of any order. Based on this, we found deterministic samples of the azimuth density, which capture its skewness and multi-modal nature. Application to range-only filtering is a direct consequence. Comparison with quadrature shows quick convergence of the formulation.

### 6.0.1 Future Work

The solutions proposed to the underlying problems in this thesis are novel and developed from first principles. Several extensions to these methods are possible and allude to extensive study. The author suggests the following directions for further research for which this thesis can serve as a foundation :

- i)* Like generalized covariance intersection, the proposed HMD can be implemented to a fusion of multi-objective probability density functions arising in the random finite set framework. Such implementations are widely used in single-sensor filtering and form the basis of multi-target tracking and extended-target tracking. The Gaussian mixture formulation, like GM-PHD filters, can be easily fused using the proposed method.

- ii)* Such an implementation would require the derivation of Pearson  $\chi^2$  divergence between multi-objective probability densities. The author believes the divergence will always exist in such cases.
- iii)* A generalization of the fusion weight  $\omega$  is not present in the literature, and all conservative fusion strategies attempt to minimize the trace or determinant of the fused covariance. Such generalization would allow non-Gaussian densities to be fused without the use of any heuristics. A similar result would be required in the case of sampling-based methods.
- iv)* In covariance reconstruction at local node, a vector of local measurement update times was mandatory. The paper hints at how to resolve this without any such information, but the technique is not tested and implemented. Further verification would ease the communication load even more, in tighter situations.
- v)* The covariance reconstruction mechanism can be generalized to more motion models other than NCA and NCV. Similarly, closed-form expressions for various sensor models employed in the tracking domain could be derived.
- vi)* The calculation of deterministic points based on circular moments has not been fine-tuned. More efficient optimization-based algorithms are needed.
- vii)* Calculation of modified Bessel function ratios with respect to order 0.

# Appendix A

## Proof that the division of Gaussian densities performed in equation (2.6.5) is always valid

The numerator, which is a valid product density has the covariance,

$$\begin{aligned}\mathbf{\Gamma}^{num} &= \left( \mathbf{\Gamma}_k^{i-1} + \mathbf{\Gamma}_k^{j-1} \right)^{-1}, \\ &= \mathbf{\Gamma}_k^i - \mathbf{\Gamma}_k^i \left( \mathbf{\Gamma}_k^i + \mathbf{\Gamma}_k^j \right)^{-1} \mathbf{\Gamma}_k^i, \end{aligned} \tag{A.0.1a}$$

$$= \mathbf{\Gamma}_k^j - \mathbf{\Gamma}_k^j \left( \mathbf{\Gamma}_k^i + \mathbf{\Gamma}_k^j \right)^{-1} \mathbf{\Gamma}_k^j, \tag{A.0.1b}$$

where the matrix inversion lemma [14] has been used. Assuming  $\mathbf{\Gamma}_k^i$  and  $\mathbf{\Gamma}_k^j$  are invertible, equation (A.0.1) implies that in the sense of positive definiteness,  $\mathbf{\Gamma}^{num} \preceq$

$\{\Gamma_k^i, \Gamma_k^j\}$ . Moreover, for the division to be valid, it is required that [2],

$$\Gamma_k^{eq} \succeq \Gamma_k^{num}, \quad \forall k \quad (\text{A.0.2})$$

Using the property of arithmetic average,

$$\Gamma_k^{eq} \succ \omega \Gamma_k^i + (1 - \omega) \Gamma_k^j \succeq \min\{\Gamma_k^i, \Gamma_k^j\}, \quad (\text{A.0.3})$$

we reach  $\Gamma_k^{eq} - \Gamma_k^{num} \succ \mathbf{0}$ ,  $\forall k$  which concludes the proof.

# Appendix B

## Proof that reported covariance in ICI is greater than in HMD-GA

Indirectly, it is sufficient to prove that all eigen values of the covariance in ICI are greater than that in HMD-GA. For notation, the eigen values of an n-dimensional matrix  $\mathbf{A}$  are represented by  $\lambda_A^1, \lambda_A^2, \dots, \lambda_A^n$  with  $\lambda_A^1 \geq \lambda_A^2, \dots \geq \lambda_A^n$ .

It's easy to see that interchange of fusion weight  $\omega$  in the formulation of both HMD-GA and ICI has no significant effect since it hovers around 0.3 – 0.6 in most pragmatic cases. Thus, we can perceive  $\mathbf{\Gamma}_m^{\text{HMD}}$  as  $\mathbf{\Gamma}_m^{\text{ICI}} + \omega(1 - \omega)\tilde{\mathbf{\Gamma}}$ , where  $\tilde{\mathbf{\Gamma}}$  is the spread-of-means term. The inverse of mutual information component in HMD-GA is (using Sherman-Morrison Formula [44]),

$$(\mathbf{\Gamma}_m^{\text{HMD}})^{-1} = (\mathbf{\Gamma}_m^{\text{ICI}})^{-1} - \omega(1 - \omega) \frac{(\mathbf{\Gamma}_m^{\text{ICI}})^{-1} \tilde{\mathbf{\Gamma}} (\mathbf{\Gamma}_m^{\text{ICI}})^{-1}}{1 + \bar{\mathbf{x}}^T (\mathbf{\Gamma}_m^{\text{ICI}})^{-1} \bar{\mathbf{x}}} \quad (\text{B.0.1})$$

where  $\bar{\mathbf{x}} = (\hat{\mathbf{x}}^i - \hat{\mathbf{x}}^j)$ . This leads to following relation between fused covariance in



the case of HMD-GA and ICI,

$$\mathbf{\Gamma}_{\text{HMD}}^f = \left[ \left( \mathbf{\Gamma}_{\text{ICI}}^f \right)^{-1} + \frac{\left( \mathbf{\Gamma}_m^{\text{ICI}} \right)^{-1} \tilde{\mathbf{\Gamma}} \left( \mathbf{\Gamma}_m^{\text{ICI}} \right)^{-1}}{1 + \bar{\mathbf{x}}^T \left( \mathbf{\Gamma}_m^{\text{ICI}} \right)^{-1} \bar{\mathbf{x}}} \right]^{-1} \quad (\text{B.0.2})$$

Again, using rank-one update of an inverse,

$$\mathbf{\Gamma}_{\text{HMD}}^f = \mathbf{\Gamma}_{\text{ICI}}^f - \omega(1 - \omega) \frac{1}{1 + \bar{\mathbf{x}}^T \left( \mathbf{\Gamma}_m^{\text{ICI}} \right)^{-1} \bar{\mathbf{x}}} \times \frac{\mathbf{\Gamma}_{\text{ICI}}^f \left( \mathbf{\Gamma}_m^{\text{ICI}} \right)^{-1} \tilde{\mathbf{\Gamma}} \left( \mathbf{\Gamma}_m^{\text{ICI}} \right)^{-1} \mathbf{\Gamma}_{\text{ICI}}^f}{1 + \bar{\mathbf{y}}^T \mathbf{\Gamma}_{\text{ICI}}^f \bar{\mathbf{y}}} \quad (\text{B.0.3})$$

where  $\bar{\mathbf{y}} = \left( \mathbf{\Gamma}_m^{\text{ICI}} \right)^{-1} \bar{\mathbf{x}}$ . Note that the last term on LHS is still a rank-one matrix with at most one eigen value  $\lambda^{\text{SOM}}$ , since it's still related to spread-of-means (SOM) term.

Using eqn. (B.0.3), the eigen values should follow the relation,

$$\lambda_1^{\text{HMD}} + \lambda_2^{\text{HMD}} + \dots = \lambda_1^{\text{ICI}} + \lambda_2^{\text{ICI}} + \dots + \lambda^{\text{SOM}}. \quad (\text{B.0.4})$$

Relations between individual eigen values are not available in literature as per author's knowledge. The best result is Cauchy's interlacing theorem [67], which when applied to eqn. (B.0.3), results in,

$$\lambda_n^{\text{HMD}} \leq \lambda_n^{\text{ICI}} \leq \lambda_{n-1}^{\text{HMD}} \leq \lambda_{n-1}^{\text{ICI}} \dots \lambda_1^{\text{HMD}} \leq \lambda_1^{\text{ICI}} \quad (\text{B.0.5})$$

where  $n$  is the dimension of the matrices.

# Appendix C

## Calculation of Fudge Factor in equation (4.5.5)

First, we propose the following upper bound for  $\sigma_{\theta_t}^2$  in equation (4.5.4b).

$$\begin{aligned}\sigma_{\theta_t}^2 &\leq \frac{\delta(x) x_{k|k}^2 + \delta(y) y_{k|k}^2}{(x_{k|k}^2 + y_{k|k}^2)^2} \\ &\leq \frac{\max(\delta(x), \delta(y))}{(x_{k|k}^2 + y_{k|k}^2)} \\ &\leq \frac{\sigma_{r_t}^2}{r_t^2}\end{aligned}\tag{C.0.1}$$

Rewriting equations (4.5.2a) through (4.5.2c) as

$$\begin{aligned}R_k^t(1, 1) &= \bar{R}_k^t(1, 1) + f(\sigma_{r_t}, \sigma_{\theta_t}) \approx \eta \bar{R}_k^t(1, 1) \\ R_k^t(2, 2) &= \bar{R}_k^t(2, 2) + g(\sigma_{r_t}, \sigma_{\theta_t}) \approx \eta \bar{R}_k^t(2, 2)\end{aligned}$$

where  $\bar{R}_k^t(1, 1)$  is the term obtained by setting  $\sigma_{r_t} = 0$  and,

$$\begin{aligned} f(\cdot) &= \frac{1}{2}\sigma_{r_t}^2 \left[ 1 + \cos(2\theta_t)e^{-2\sigma_\theta^2}e^{-2\sigma_{\theta_t}^2} \right] e^{\sigma_\theta^2} \\ &\quad - \frac{1}{2}\sigma_{r_t}^2 \left[ 1 + \cos(2\theta_t)e^{-2\sigma_{\theta_t}^2} \right] = \bar{R}_k^t(1, 1)(\eta - 1) \end{aligned} \quad (\text{C.0.2a})$$

$$\begin{aligned} g(\cdot) &= \frac{1}{2}\sigma_{r_t}^2 \left[ 1 - \cos(2\theta_t)e^{-2\sigma_\theta^2}e^{-2\sigma_{\theta_t}^2} \right] e^{\sigma_\theta^2} \\ &\quad - \frac{1}{2}\sigma_{r_t}^2 \left[ 1 - \cos(2\theta_t)e^{-2\sigma_{\theta_t}^2} \right] = \bar{R}_k^t(2, 2)(\eta - 1) \end{aligned} \quad (\text{C.0.2b})$$

It can be seen that,

$$\begin{aligned} -\frac{f(\cdot) - g(\cdot)}{f(\cdot) + g(\cdot)} &= e^{-2\sigma_{\theta_t}^2} \cos(2\theta_t) \approx 1 - 2\sigma_{\theta_t}^2 \\ &\approx 1 - 2\frac{\sigma_{r_t}^2}{r_t^2} = 1 - 2\frac{f(\cdot) + g(\cdot)}{r_t^2(e^{\sigma_\theta^2} - 1)} \end{aligned} \quad (\text{C.0.3})$$

Using equations (C.0.2a) and (C.0.2b), equation (C.0.3) can be converted to a linear equation in  $\eta$  where rest of the parameters are known which can be solved, resulting in,

$$\begin{aligned} \eta &= \left[ 1 - \frac{\bar{R}^t(1, 1) - \bar{R}^t(2, 2)}{\bar{R}^t(1, 1) + \bar{R}^t(2, 2)} \frac{(e^{\sigma_\theta^2} - 1)}{\cos(2\theta_t)(e^{-\sigma_\theta^2} - 1)} \right] \\ &\quad \times \left[ \frac{r_t^2(e^{\sigma_\theta^2} - 1)}{2\{\bar{R}^t(1, 1) + \bar{R}^t(2, 2)\}} \right] + 1 \end{aligned} \quad (\text{C.0.4})$$

# Appendix D

## Second Order Linearization of CMM in equation (4.6.8)

Eqn. (4.6.2) states,

$$\mathbf{g}(\mathbf{z}_k - \mathbf{v}_k) = \mathbf{x}_k \tag{D.0.1}$$

Using second-order Taylor series approximation about the expected measurement  $\hat{\mathbf{z}}_{k|k} = \mathbb{E}(\mathbf{z}_k | \hat{\mathbf{x}}_{k|k})$ , (omitting  $k$ ),

$$\begin{aligned} \mathbf{g}(\mathbf{z} - \mathbf{v}) &\approx \mathbf{g}(\hat{\mathbf{z}}) + \nabla_{\hat{\mathbf{z}}}^{\mathbf{z}}(\mathbf{z} - \hat{\mathbf{z}} - \mathbf{v}) + \frac{1}{2} \begin{bmatrix} \vdots \\ (\mathbf{z} - \hat{\mathbf{z}} - \mathbf{v})^T \mathcal{H}_i^{\hat{\mathbf{z}}}(\mathbf{z} - \hat{\mathbf{z}} - \mathbf{v}) \\ \vdots \end{bmatrix} \\ &= \mathbf{x} \end{aligned} \tag{D.0.2}$$

where  $\mathcal{H}_i^{\hat{\mathbf{z}}}$  is the Hessian matrix corresponding to the  $i^{\text{th}}$  element of the vector valued function  $\mathbf{g}(\cdot)$  evaluated at  $\hat{\mathbf{z}}$ . Separating the terms containing noise variable  $\mathbf{v}$  and assuming  $\mathbf{z} - \hat{\mathbf{z}} \approx \mathbf{v}$ ,

$$\begin{aligned} \mathbf{z}^m &= \mathbf{g}(\hat{\mathbf{z}}) + \nabla_g^{\hat{\mathbf{z}}}(\mathbf{z} - \hat{\mathbf{z}}) + \frac{1}{2} \begin{bmatrix} \vdots \\ (\mathbf{z} - \hat{\mathbf{z}})^T \mathcal{H}_i^{\hat{\mathbf{z}}}(\mathbf{z} - \hat{\mathbf{z}}) \\ \vdots \end{bmatrix} \\ &= \mathbf{x} + \underbrace{\nabla_g^{\hat{\mathbf{z}}}\mathbf{v} + \frac{1}{2} \begin{bmatrix} \vdots \\ \mathbf{v}^T \mathcal{H}_i^{\hat{\mathbf{z}}}\mathbf{v} \\ \vdots \end{bmatrix}}_{\text{converted measurement noise}} \end{aligned} \quad (\text{D.0.3})$$

The remaining part is to calculate the covariance of the converted measurement noise in R.H.S of equation (D.0.3). Following properties are worth mentioning,

- Since  $\mathbf{v}$  is zero mean,

$$\mathbb{E}(\mathbf{v}^T \mathcal{H}_i^{\hat{\mathbf{z}}} \mathbf{v}) = \text{tr}(\mathcal{H}_i^{\hat{\mathbf{z}}} \mathbf{R}) \quad (\text{D.0.4})$$

- Since  $\mathbf{v}$  is assumed Gaussian and zero mean,

$$\mathbb{E}(\nabla_g^{\hat{\mathbf{z}}}\mathbf{v}\mathbf{v}^T \mathcal{H}_i^{\hat{\mathbf{z}}}\mathbf{v}) = \mathbf{0} \quad (\text{D.0.5})$$

- Covariance of  $\mathbf{v}$  is  $\mathbf{R}$ , hence

$$\text{cov}(\nabla_g^{\hat{\mathbf{z}}}\mathbf{v}) = \nabla_g^{\hat{\mathbf{z}}}\mathbf{R}\nabla_g^{\hat{\mathbf{z}}T} \quad (\text{D.0.6})$$

Using these properties, the calculation of noise covariance in equation (D.0.3) is straightforward but tedious, the result is,

$$\mathbf{R}^g = \frac{1}{4} \begin{bmatrix} 2 \text{tr}(\mathcal{H}_1^{\hat{\mathbf{z}}}\mathbf{R}\mathcal{H}_1^{\hat{\mathbf{z}}}\mathbf{R}) & \cdots & 2 \text{tr}(\mathcal{H}_1^{\hat{\mathbf{z}}}\mathbf{R}\mathcal{H}_{n_x}^{\hat{\mathbf{z}}}\mathbf{R}) \\ \vdots & \ddots & \vdots \\ 2 \text{tr}(\mathcal{H}_{n_x}^{\hat{\mathbf{z}}}\mathbf{R}\mathcal{H}_1^{\hat{\mathbf{z}}}\mathbf{R}) & \cdots & 2 \text{tr}(\mathcal{H}_{n_x}^{\hat{\mathbf{z}}}\mathbf{R}\mathcal{H}_{n_x}^{\hat{\mathbf{z}}}\mathbf{R}) \end{bmatrix} + \nabla_g^{\hat{\mathbf{z}}}\mathbf{R}\nabla_g^{\hat{\mathbf{z}}T} \quad (\text{D.0.7})$$

Note that the converted measurement in equation (D.0.3) contains an additive bias which is equal to the mean of converted noise on R.H.S.

# Appendix E

## Proof of Eqns. (5.6.4) and (5.6.5)

We wish to compute the integral  $I_{\text{diag,cos}}(m)$ ,

$$\int_0^{2\pi} \exp \left[ r_k \|p, q\| \cos(\theta_k - \phi_1) + A_3 r_k^2 \cos(2\theta_k) \right] \cos(m\theta_k) d\theta_k \quad (\text{E.0.1})$$

which can be rewritten in the standard form as,

$$\int_0^{2\pi} \exp \left[ A_1 r_k \cos(\theta_k) + A_2 r_k \sin(\theta_k) + A_3 r_k^2 \cos(2\theta_k) \right] \cos(m\theta_k) d\theta_k \quad (\text{E.0.2})$$

where  $A_1$  and  $A_2$  are given in eqn. (5.5.7). Substituting the following identities for the Bessel function in the equation above,

$$\exp [z \sin(\theta)] = \sum_{n=-\infty}^{\infty} I_n(z) \exp \left[ ni \left( \theta + \frac{3\pi}{2} \right) \right] \quad (\text{E.0.3})$$

$$\exp [z \cos(\theta)] = \sum_{n=-\infty}^{\infty} I_n(z) \exp [ni\theta] \quad (\text{E.0.4})$$

where  $I_n(z)$  is the modified Bessel function of the first kind and order  $n$ . The result is (removing the subscript  $k$  for time),

$$\begin{aligned} I_{\text{diag,cos}}(m) &= \int_0^{2\pi} \cos(m\theta) d\theta \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} I_j(A_3 r^2) \\ &\quad \times I_k(A_1 r) I_l(A_2 r) \exp \left[ i\theta_k \underbrace{(2j + k + l)}_M + i \underbrace{\frac{3\pi l}{2}}_L \right] \end{aligned} \quad (\text{E.0.5})$$

Moving the integral inside the summation, and using Euler's identity for  $\cos(m\theta)$ ,

$$\begin{aligned} I_{\text{diag,cos}}(m) &= \sum_j \sum_k \sum_l \frac{1}{2} I_j(A_3 r^2) I_k(A_1 r) I_l(A_2 r) \times \\ &\quad \int_0^{2\pi} \left[ \exp(i\theta(M + m) + iL) + \exp(i\theta_k(M - m) + iL) \right] d\theta \end{aligned} \quad (\text{E.0.6})$$

where  $M$  and  $L$  are indicated in eqn. (E.0.5). Taking the real part of the above equation and using the following identities for an integer  $m$ ,

$$\int_0^{2\pi} \cos(m\theta) = \begin{cases} 2\pi & m = 0 \\ 0 & \text{otherwise} \end{cases} \quad (\text{E.0.7a})$$

$$\int_0^{2\pi} \sin(m\theta) = 0 \quad (\text{E.0.7b})$$

which means that the integral is non-zero only for following cases,

$$(M \pm m) = 0, \quad l = 2n \quad (\text{E.0.8})$$



The resulting integral is then,

$$\begin{aligned} I_{\text{diag,cos}}(m) = & \frac{2\pi}{2} \sum_j I_j(A_3 r^2) \sum_{n=-\infty}^{\infty} (-1)^n \left[ I_{2n+2j+m}(A_1 r) I_{2n}(A_2 r) \right. \\ & \left. + I_{2n+2j-m}(A_1 r) I_{2n}(A_2 r) \right] \end{aligned} \quad (\text{E.0.9})$$

Now, use of the following addition theorems [102] is made which reduces the above expression to a single sum.

$$\sum_{q=-\infty}^{\infty} (-1)^q I_{p+2q}(Z) I_{2q}(z) = I_p \left( \sqrt{Z^2 + z^2} \right) \cos(p\psi), \quad (\text{E.0.10})$$

where  $\tan(\psi) = z/Z$ . Eqn. (5.6.2) follows from the substitution of eqn. (E.0.10) in (E.0.9).

The proof for  $I_{\text{diag,sin}}(m)$  follows in a similar manner except that Euler identity for sin is used in eqn. (E.0.6),

$$\begin{aligned} I_{\text{diag,sin}}(m) = & \sum_j \sum_k \sum_l \frac{1}{2} I_j(A_3 r^2) I_k(A_1 r) I_l(A_2 r) \int_0^{2\pi} \left[ \exp \left( i\theta(M + m) + i(L - \frac{\pi}{2}) \right) \right. \\ & \left. - \exp \left( i\theta_k(M - m) + i(L - \frac{\pi}{2}) \right) \right] d\theta \end{aligned} \quad (\text{E.0.11})$$

The integral is non-zero for the following cases,

$$(M \pm m) = 0, \quad l = 2n + 1 \quad (\text{E.0.12})$$

Making use of the identities in eqn. (E.0.7), the integral reduces to,

$$\begin{aligned} I_{\text{diag, sin}}(m) = \frac{2\pi}{2} \sum_j I_j(A_3 r^2) \sum_{n=-\infty}^{\infty} (-1)^{n+1} \left[ I_{2n+2j+m+1}(A_1 r) I_{2n+1}(A_2 r) \right. \\ \left. - I_{2n+2j-m+1}(A_1 r) I_{2n+1}(A_2 r) \right] \end{aligned} \quad (\text{E.0.13})$$

Again, we make use of the following addition theorem which reduces the expression to a single summation,

$$\sum_{q=-\infty}^{\infty} (-1)^{q+1} I_{p+2q+1}(Z) I_{2q+1}(z) = I_p \left( \sqrt{Z^2 + z^2} \right) \sin(p\psi). \quad (\text{E.0.14})$$

Equation (5.6.3) is the result of substitution of eqn. (E.0.14) in eqn. (E.0.13).

# Bibliography

- [1] Abbas, A. E. (2009). A Kullback-Leibler view of linear and log-linear pools. *Decision Analysis*, **6**(1), 25–37.
- [2] Acar, D. and Orguner, U. (2020). Decorrelation of previously communicated information for an interacting multiple model filter. *IEEE Transactions on Aerospace and Electronic Systems*.
- [3] Ahmed, N. R. (2015). What’s one mixture divided by another?: A unified approach to high-fidelity distributed data fusion with mixture models. In *2015 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 289–296. IEEE.
- [4] Ajgl, J. and Šimandl, M. (2014). Conservativeness of estimates given by probability density functions: Formulation and aspects. *Information Fusion*, **20**, 117–128.
- [5] Ajgl, J., Šimandl, M., and Duník, J. (2015). Approximation of powers of Gaussian mixtures. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 878–885. IEEE.
- [6] Alspach, D. and Sorenson, H. (1972). Nonlinear bayesian estimation using Gaussian sum approximations. *IEEE transactions on automatic control*, **17**(4), 439–448.

- 
- [7] Amos, D. E. (1974). Computation of modified bessel functions and their ratios. *Mathematics of computation*, **28**(125), 239–251.
- [8] Bailey, T., Julier, S., and Agamennoni, G. (2012). On conservative fusion of information with unknown non-Gaussian dependence. In *2012 15th International Conference on Information Fusion*, pages 1876–1883. IEEE.
- [9] Bar-Shalom, Y. (1981). On the track-to-track correlation problem. *IEEE Transactions on Automatic control*, **26**(2), 571–572.
- [10] Bar-Shalom, Y. and Campo, L. (1986). The effect of the common process noise on the two-sensor fused-track covariance. *IEEE Transactions on aerospace and electronic systems*, (6), 803–805.
- [11] Bar-Shalom, Y. and Chen, H. (2008). Covariance reconstruction for track fusion with legacy track sources. *J. Adv. Inf. Fusion*, **3**(2), 107–117.
- [12] Bar-Shalom, Y. and Li, X.-R. (1995). *Multitarget-Multisensor Tracking: Principles and Techniques*, volume 19. YBs Storrs, CT.
- [13] Bar-Shalom, Y., Li, X. R., and Kirubarajan, T. (2001). *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons.
- [14] Bar-Shalom, Y., Li, X. R., and Kirubarajan, T. (2004). *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons.
- [15] Battistelli, G. and Chisci, L. (2014). Kullback–Leibler average, consensus on

- probability densities, and distributed state estimation with guaranteed stability. *Automatica*, **50**(3), 707–718.
- [16] Best, R. A. and Norton, J. (1997). A new model and efficient tracker for a target with curvilinear motion. *IEEE Transactions on Aerospace and Electronic Systems*, **33**(3), 1030–1037.
- [17] Bhaumik, S. and Date, P. (2019). *Nonlinear estimation: methods and applications with deterministic Sample Points*. CRC Press.
- [18] Bibby, J. (1974). Axiomatisations of the average and a further generalisation of monotonic sequences. *Glasgow Mathematical Journal*, **15**(1), 63–65.
- [19] Blackman, S. S. (1986). Multiple-target tracking with radar applications. *Dedham*.
- [20] Bordonaro, S., Willett, P., and Bar-Shalom, Y. (2014). Decorrelated unbiased converted measurement kalman filter. *IEEE Transactions on Aerospace and Electronic Systems*, **50**(2), 1431–1444.
- [21] Bordonaro, S. V. (2015). Converted measurement trackers for systems with nonlinear measurement functions.
- [22] Bordonaro, S. V., Willett, P., and Bar-Shalom, Y. (2012). Unbiased tracking with converted measurements. In *2012 IEEE Radar Conference*, pages 0741–0745. IEEE.
- [23] Chang, K.-C., Saha, R. K., and Bar-Shalom, Y. (1997). On optimal track-to-track fusion. *IEEE Transactions on Aerospace and Electronic Systems*, **33**(4), 1271–1276.

- [24] Chang, K. C., Zhi, T., and Saha, R. K. (2002). Performance evaluation of track fusion with information matrix filter. *IEEE Transactions on Aerospace and Electronic Systems*, **38**(2), 455–466.
- [25] Chang, K. C., Chong, C. Y., and Mori, S. (2008). On scalable distributed sensor fusion. In *2008 11th International Conference on Information Fusion*, pages 1–8. IEEE.
- [26] Chang, K.-C., Chong, C.-Y., and Mori, S. (2010). Analytical and computational evaluation of scalable distributed fusion algorithms. *IEEE transactions on Aerospace and Electronic Systems*, **46**(4), 2022–2034.
- [27] Chen, L., Arambel, P. O., and Mehra, R. K. (2002). Fusion under unknown correlation-covariance intersection as a special case. In *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002. (IEEE Cat. No. 02EX5997)*, volume 2, pages 905–912. IEEE.
- [28] Civicioglu, P. (2012). Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm. *Computers & Geosciences*, **46**, 229–247.
- [29] Clark, D. E., Panta, K., and Vo, B.-N. (2006). The GM-PHD filter multiple target tracker. In *2006 9th International Conference on Information Fusion*, pages 1–8. IEEE.
- [30] Clark, J. M., Kountouriotis, P.-A., and Vinter, R. B. (2009). Gaussian mixture filtering for range only tracking problems. In *2009 European Control Conference (ECC)*, pages 49–54. IEEE.

- [31] Clark, J. M., Kountouriotis, P.-A., and Vinter, R. B. (2010). A gaussian mixture filter for range-only tracking. *IEEE transactions on automatic control*, **56**(3), 602–613.
- [32] Cover, T. M. (1999). *Elements of information theory*. John Wiley & Sons.
- [33] Csiszár, I., Shields, P. C., *et al.* (2004). Information theory and statistics: A tutorial. *Foundations and Trends<sup>®</sup> in Communications and Information Theory*, **1**(4), 417–528.
- [34] Da, K., Li, T., Zhu, Y., Fan, H., and Fu, Q. (2019). Kullback-Leibler averaging for multitarget density fusion. In *International Symposium on Distributed Computing and Artificial Intelligence*, pages 253–261. Springer.
- [35] de Carvalho, M. (2016). Mean, what do you mean? *The American Statistician*, **70**(3), 270–274.
- [36] Dell-Imagine, R. (1976). Jtids- an overview of the system design and implementation(joint tactical information distribution system). In *Position Location and Navigation Symposium, San Diego, Calif*, pages 212–215.
- [37] Drummond, O. E. (1997). Tracklets and a hybrid fusion with process noise. In *Signal and Data Processing of Small Targets 1997*, volume 3163, pages 512–524. International Society for Optics and Photonics.
- [38] Drummond, O. E. (2002). Track and tracklet fusion filtering. In *Signal and Data Processing of Small Targets 2002*, volume 4728, pages 176–195. International Society for Optics and Photonics.

- [39] Gatto, R. and Jammalamadaka, S. R. (2007). The generalized von mises distribution. *Statistical Methodology*, **4**(3), 341–353.
- [40] Govaers, F. and Koch, W. (2011). On the globalized likelihood function for exact track-to-track fusion at arbitrary instants of time. In *14th International Conference on Information Fusion*, pages 1–5. IEEE.
- [41] Govaers, F. and Koch, W. (2012). An exact solution to track-to-track-fusion at arbitrary communication rates. *IEEE Transactions on Aerospace and Electronic Systems*, **48**(3), 2718–2729.
- [42] Govaers, F., Charlish, A., and Koch, W. (2013). Distributed tracking with constrained communication. In *2013 Military Communications and Information Systems Conference*, pages 1–5. IEEE.
- [43] Gunay, M., Orguner, U., and Demirekler, M. (2016). Chernoff fusion of Gaussian mixtures based on sigma-point approximation. *IEEE Transactions on Aerospace and Electronic Systems*, **52**(6), 2732–2746.
- [44] Hager, W. W. (1989). Updating the inverse of a matrix. *SIAM review*, **31**(2), 221–239.
- [45] Hanebeck, U. D. and Lindquist, A. (2014). Moment-based dirac mixture approximation of circular densities. *IFAC Proceedings Volumes*, **47**(3), 5040–5048.
- [46] Higham, N. J. (1992). Estimating the matrix p-norm. *Numerische Mathematik*, **62**(1), 539–555.



- [47] Hurley, M. B. (2002). An information theoretic justification for covariance intersection and its generalization. In *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002. (IEEE Cat. No. 02EX5997)*, volume 1, pages 505–511. IEEE.
- [48] Julier, S. (2008). Fusion without independence. *IET Conference Proceedings*, pages 1–4(3).
- [49] Julier, S. J. (2006). An empirical study into the use of Chernoff information for robust, distributed fusion of Gaussian mixture models. In *2006 9th International Conference on Information Fusion*, pages 1–8. IEEE.
- [50] Julier, S. J. and Uhlmann, J. K. (1997). A non-divergent estimation algorithm in the presence of unknown correlations. In *Proceedings of the 1997 American Control Conference (Cat. No. 97CH36041)*, volume 4, pages 2369–2373. IEEE.
- [51] Kan, L., Zhang, Y., Fan, H., Yang, W., and Chen, Z. (2008). Matlab-based simulation of buoyancy-driven underwater glider motion. *Journal of Ocean University of China*, **7**(1), 113–118.
- [52] Kantor, G. and Singh, S. (2002). Preliminary results in range-only localization and mapping. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 2, pages 1818–1823. Ieee.
- [53] Kirubarajan, T. and Bar-Shalom, Y. (2017). Target tracking using probabilistic data association-based techniques with applications to sonar, radar, and eo sensors. In *Handbook of multisensor data fusion*, pages 223–262. CRC Press.

- [54] Koliander, G., El-Laham, Y., Djurić, P. M., and Hlawatsch, F. (2022). Fusion of probability density functions. *Proceedings of the IEEE*, **110**(4), 404–453.
- [55] Kolmogorov, A. (1991). Selected works of A.N. Kolmogorov. vol. i: Mathematics and mechanics. edited by Tikhomirov V.M.
- [56] Kou, S. R., Elliott, D. L., and Tarn, T. J. (1973). Observability of nonlinear systems. *Information and Control*, **22**(1), 89–99.
- [57] Kumar, M. (1988). World geodetic system 1984: A modern and accurate global reference frame. *Marine Geodesy*, **12**(2), 117–126.
- [58] Kurz, G., Gilitschenski, I., Siegart, R. Y., and Hanebeck, U. (2016). Methods for deterministic approximation of circular densities. *Journal of Advances in Information Fusion*, **11**(2), 138–156.
- [59] Lerro, D. and Bar-Shalom, Y. (1993). Tracking with debiased consistent converted measurements versus EKF. *IEEE transactions on aerospace and electronic systems*, **29**(3), 1015–1022.
- [60] Li, T., Wang, X., Liang, Y., and Pan, Q. (2020). On arithmetic average fusion and its application for distributed multi-Bernoulli multitarget tracking. *IEEE Transactions on Signal Processing*, **68**, 2883–2896.
- [61] Li, X. R., Zhu, Y., Wang, J., and Han, C. (2003). Optimal linear estimation fusion. i. unified fusion rules. *IEEE Transactions on Information Theory*, **49**(9), 2192–2208.

- [62] Lobbia, R. N. and Owen, M. (1998). Data fusion within a constrained communication environment. Technical report, BOEING INFORMATION SPACE AND DEFENSE SYSTEMS SEATTLE WA.
- [63] Longbin, M., Xiaoquan, S., Yiyu, Z., Kang, S. Z., and Bar-Shalom, Y. (1998). Unbiased converted measurements for tracking. *IEEE Transactions on Aerospace and Electronic Systems*, **34**(3), 1023–1027.
- [64] Lu, K., Sun, C., Fu, Q., and Zhu, Q. (2019). Distributed track-to-track fusion for non-linear systems with Gaussian mixture noise. *IET Radar, Sonar & Navigation*, **13**(5), 740–749.
- [65] Mahler, R. P. (2000). Optimal/robust distributed data fusion: a unified approach. In *Signal Processing, Sensor Fusion, and Target Recognition IX*, volume 4052, pages 128–138. International Society for Optics and Photonics.
- [66] McLachlan, G. J., Lee, S. X., and Rathnayake, S. I. (2019). Finite mixture models. *Annual review of statistics and its application*, **6**, 355–378.
- [67] Mercer, A. M., Mercer, P. R., *et al.* (2000). Cauchy’s interlace theorem and lower bounds for the spectral radius. *International Journal of Mathematics and Mathematical Sciences*, **23**, 563–566.
- [68] Niculescu, C. and Persson, L.-E. (2006). *Convex functions and their applications*. Springer.
- [69] Nielsen, F. (2019). On the Jensen–Shannon symmetrization of distances relying on abstract means. *Entropy*, **21**(5), 485.
- [70] Nielsen, F. and Bhatia, R. (2013). *Matrix information geometry*. Springer.

- [71] Nielsen, F. and Garcia, V. (2009). Statistical exponential families: A digest with flash cards. *arXiv preprint arXiv:0911.4863*.
- [72] Noack, B., Reinhardt, M., and Hanebeck, U. D. (2014). On nonlinear track-to-track fusion with Gaussian mixtures. In *17th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE.
- [73] Noack, B., Sijs, J., Reinhardt, M., and Hanebeck, U. D. (2017a). Decentralized data fusion with inverse covariance intersection. *Automatica*, **79**, 35–41.
- [74] Noack, B., Sijs, J., and Hanebeck, U. D. (2017b). Inverse covariance intersection: New insights and properties. In *2017 20th International Conference on Information Fusion (Fusion)*, pages 1–8. IEEE.
- [75] Nocedal, J. and Wright, S. J. (1999). *Numerical optimization*. Springer.
- [76] Olver, F. W., Lozier, D. W., Boisvert, R. F., and Clark, C. W. (2010). *NIST handbook of mathematical functions hardback and CD-ROM*. Cambridge university press.
- [77] Olver, F. W. J., Daalhuis, A. B. O., Lozier, D. W., Schneider, B. I., Boisvert, R. F., Clark, C. W., B. R. Mille and, B. V. S., Cohl, H. S., and M. A. McClain, e. (2020). Nist digital library of mathematical functions. <http://dlmf.nist.gov/>, Release 1.0.26 of 2020-03-15.
- [78] Pillon, D., Perez-Pignol, A.-C., and Jauffret, C. (2016). Observability: range-only vs. bearings-only target motion analysis for a leg-by-leg observer’s trajectory. *IEEE Transactions on Aerospace and Electronic Systems*, **52**(4), 1667–1678.

- [79] Rahmathullah, A. S., García-Fernández, Á. F., and Svensson, L. (2017). Generalized optimal sub-pattern assignment metric. In *2017 20th International Conference on Information Fusion (Fusion)*, pages 1–8. IEEE.
- [80] Reinhardt, M., Noack, B., and Hanebeck, U. D. (2012). Closed-form optimization of covariance intersection for low-dimensional matrices. In *2012 15th International Conference on Information Fusion*, pages 1891–1896. IEEE.
- [81] Rice, S. O. (1944). Mathematical analysis of random noise. *The Bell System Technical Journal*, **23**(3), 282–332.
- [82] Ristic, B., Arulampalam, S., and McCarthy, J. (2002). Target motion analysis using range-only measurements: algorithms, performance and application to isar data. *Signal Processing*, **82**(2), 273–296.
- [83] Shampine, L. F. (2008). Vectorized adaptive quadrature in matlab. *Journal of Computational and Applied Mathematics*, **211**(2), 131–140.
- [84] Sharma, N., Tharmarasa, R., Bhaumik, S., and Kirubarajan, T. (2022). On conservative track fusion strategies : Harmonic mean density (submitted). *IEEE Transactions on Aerospace and Electronic Systems*.
- [85] Sijs, J., Lazar, M., and Bosch, P. (2010). State fusion with unknown correlation: Ellipsoidal intersection. In *Proceedings of the 2010 American Control Conference*, pages 3992–3997. IEEE.
- [86] Sirichai, P. and Yamakita, M. (2013). Using ensemble Kalman filter for distributed sensor fusion. *Transactions of the Institute of Systems, Control and Information Engineers*, **26**(12), 466–476.

- [87] Smith, A. (1982). Classroom notes: A demonstration program to illustrate the definition of bijection. *International Journal of Mathematical Education in Science and Technology*, **13**(5), 641–660.
- [88] Song, T. L. (1999). Observability of target tracking with range-only measurements. *IEEE Journal of Oceanic Engineering*, **24**(3), 383–387.
- [89] Tian, X. (2010). *Tracking, Track-to-Track Fusion and Surveillance*. Ph.D. thesis, University of Connecticut.
- [90] Tian, X. and Bar-Shalom, Y. (2008). Sequential track-to-track fusion algorithm: exact solution and approximate implementation. In *Signal and Data Processing of Small Targets 2008*, volume 6969, pages 325–336. SPIE.
- [91] Tian, X. and Bar-Shalom, Y. (2009). Exact algorithms for four track-to-track fusion configurations: All you wanted to know but were afraid to ask. In *2009 12th International Conference on Information Fusion*, pages 537–544. IEEE.
- [92] Tian, X. and Bar-Shalom, Y. (2010). On algorithms for asynchronous track-to-track fusion. In *2010 13th International Conference on Information Fusion*, pages 1–8. IEEE.
- [93] Tian, X. and Bar-Shalom, Y. (2014). A consistency-based gaussian mixture filtering approach for the contact lens problem. *IEEE Transactions on Aerospace and Electronic Systems*, **50**(3), 1636–1646.
- [94] Tian, X., Yuan, T., and Bar-Shalom, Y. (2012). Track-to-track fusion in linear and nonlinear systems. In *Itzhack Y. Bar-Itzhack Memorial Symposium on Estimation, Navigation, and Spacecraft Control*, pages 21–41. Springer.

- [95] Tichavsky, P., Muravchik, C. H., and Nehorai, A. (1998). Posterior cramer-rao bounds for discrete-time nonlinear filtering. *IEEE Transactions on signal processing*, **46**(5), 1386–1396.
- [96] tired (<https://math.stackexchange.com/users/101233/tired>) (2015). Ratio of modified bessel functions of the first kind of nth and 0th order. Mathematics Stack Exchange. URL:<https://math.stackexchange.com/q/1495410> (version: 2015-10-24).
- [97] Tucker, W. (1999). The lorenz attractor exists. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, **328**(12), 1197–1202.
- [98] Uney, M., Clark, D. E., and Julier, S. J. (2011). Information measures in distributed multitarget tracking. In *14th International Conference on Information Fusion*, pages 1–8. IEEE.
- [99] Upcroft, B., Ong, L. L., Kumar, S., Ridley, M., Bailey, T., Sukkarieh, S., and Durrant-Whyte, H. (2005). Rich probabilistic representations for bearing only decentralised data fusion. In *2005 7th International Conference on Information Fusion*, volume 2, pages 8–pp. IEEE.
- [100] Visina, R., Bar-Shalom, Y., Willett, P., and Dey, D. (2019). On-demand track-to-track fusion using local imm inside information. In *Signal Processing, Sensor/Information Fusion, and Target Recognition XXVIII*, volume 11018, page 1101804. International Society for Optics and Photonics.
- [101] Wan, E. A., Van Der Merwe, R., and Haykin, S. (2001). The unscented kalman filter. *Kalman filtering and neural networks*, **5**(2007), 221–280.

- [102] Watson, G. N. (1922). *A treatise on the theory of Bessel functions*, volume 2. The University Press.
- [103] Weil, H. (1954). The distribution of radial error. *The Annals of Mathematical Statistics*, pages 168–170.
- [104] Weisstein, E. W. (2012). Bessel function of the first kind (from mathworld—a wolfram web resource). URL: <https://mathworld.wolfram.com/BesselFunctionoftheFirstKind.html> (accessed 2020/10/15).
- [105] Yuan, T., Duraisamy, B., Schwarz, T., and Fritzsche, M. (2016). Track fusion with incomplete information for automotive smart sensor systems. In *2016 IEEE Radar Conference (RadarConf)*, pages 1–4. IEEE.
- [106] Yury, P. and Wu, Y. (2022). *Information Theory : From Coding to Learning*. Cambridge University Press, 1st edition.
- [107] Zhang, H. and Ding, F. (2013). On the kronecker products and their applications. *Journal of Applied Mathematics*, **2013**.