

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DISTRIBUTED FUSION
AND SENSOR MANAGEMENT
FOR NON-TRADITIONAL SENSOR MODELS

HASAN HÜSEYİN SÖNMEZ
A THESIS SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
DEPARTMENT OF ELECTRONICS ENGINEERING

GEBZE

2023

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF
NATURAL AND APPLIED SCIENCES

DISTRIBUTED FUSION
AND SENSOR MANAGEMENT
FOR NON-TRADITIONAL SENSOR MODELS

HASAN HÜSEYİN SÖNMEZ
A THESIS SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
DEPARTMENT OF ELECTRONICS ENGINEERING

THESIS SUPERVISOR
ASSIST. PROF. DR. ALİ KÖKSAL HOCAOĞLU

GEBZE

2023

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GELENEKSEL OLMAYAN
SENSÖR MODELLERİ İÇİN
DAĞITIK FÜZYON VE SENSÖR YÖNETİMİ

HASAN HÜSEYİN SÖNMEZ
DOKTORA TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

TEZ DANIŞMANI
DR. ÖĞR. ÜYESİ ALİ KÖKSAL HOCAOĞLU

GEBZE
2023

GTÜ Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 19/07/2023 tarih ve 2023/40 sayılı kararıyla oluşturulan jüri tarafından 15/08/2023 tarihinde tez savunma sınavı yapılan Hasan Hüseyin SÖNMEZ'in tez çalışması Elektronik Mühendisliği Anabilim Dalında DOKTORA tezi olarak kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI)

: Dr. Öğr. Üyesi Ali Köksal HOCAOĞLU

ÜYE

: Prof. Dr. Koray KAYABOL

ÜYE

: Prof. Dr. Tansu FİLİK

ÜYE

: Dr. Öğr. Üyesi Saliha BÜYÜKÇORAK EDİBALI

ÜYE

: Dr. Öğr. Üyesi Abdurrahman YILMAZ

ONAY

Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun

...../...../..... tarih ve/..... sayılı kararı.

SUMMARY

Data fusion has emerged as a crucial component in various intelligent systems, ranging from autonomous vehicles to surveillance networks. The growth in processing capabilities has garnered significant attention from researchers, facilitating the integration of diverse sensor types to achieve a more comprehensive situational understanding. Consequently, mathematical models must be tailored to different sensor models to optimize fusion processes. Moreover, the fusion procedure can be applied at multiple abstraction levels, depending on the sensor data available. Hence, the fusion algorithm should address various statistical challenges to maximize data and information exploitation.

In this dissertation, we explore the diverse abstraction levels for unified fusion systems. Our contributions to the fusion literature include unified algorithms and statistical models that can be utilized at different levels. In the context of measurement-level fusion, we devise and propose a novel dynamic clustering model, exemplifying non-traditional measurements. This model is applied to a dynamic clustering problem where cluster kinematics change in time, as well as their statistics, with the proposed algorithm, is based on Bayesian non-parametric and RFS. For estimation-level fusion, we implement a distributed fusion architecture to address the high-level sensor fusion problem. Specifically, we integrate altitude data from a radar altimeter and camera images—comprising non-homogeneous sensor information—to enhance position estimations derived from inertial sensor data. Furthermore, we introduce a novel RFS-based information functional and employ it in a sensor management problem where a mobile sensor determines its subsequent position to improve the overall information it gathers from its surroundings.

Key Words: Distributed Fusion, Sensor Management, State Estimation, Random Finite Sets.

ÖZET

Veri füzyonu, otonom araçlardan gözetleme ağlarına kadar çeşitli akıllı sistemlerde kritik bir bileşen olarak ortaya çıkmıştır. İşlem kapasitelerindeki artış, araştırmacıların önemli ölçüde ilgisini çekmiş ve daha kapsamlı bir durumsal anlayış elde etmek için çeşitli sensör türlerinin entegrasyonunu kolaylaştırmıştır. Sonuç olarak, füzyon sistemlerindeki matematiksel modeller, füzyon süreçlerinin optimizasyonu için uyarlanmalıdır. Ayrıca, füzyon işlemi, mevcut sensör verilerine bağlı olarak çeşitli soyutlama düzeylerinde uygulanabilir. Bu nedenle, füzyon algoritması, veri ve bilgi kullanımını en üst düzeye çıkarmak için çeşitli istatistiksel problemlerle başa çıkmalıdır.

Bu tezde, füzyon sistemleri farklı soyutlama düzeylerinde incelenmektedir. Füzyon literatürüne katkılarımız, füzyon mimarisinin çeşitli seviyelerinde yararlanılabilecek bütünleşik füzyon algoritmalarını ve istatistiksel modelleri içermektedir. Ölçüm düzeyinde füzyon bağlamında, geleneksel olmayan ölçümlere örnek teşkil eden yenilikçi bir dinamik kümeleme modeli önerilmektedir. Bu model, parametrik olmayan Bayesçi ve RFS temelli bir algoritma ile kümelerin hem kinematik hem de istatistiklerinin zaman içinde değiştiği dinamik kümeleme problemlerine uygulanmaktadır. Kestirim düzeyinde füzyon için, sensör füzyonu problemi dağıtık bir füzyon mimarisi ile ele alınmaktadır. Spesifik olarak, radar altimetre ve kamera görüntülerinden elde edilen konum verileri — yani homojen olmayan sensör bilgileri- ataletsel sensör verilerinden elde edilen konum tahminlerini iyileştirmek için entegre edilmektedir. Ayrıca, yeni bir RFS tabanlı enformasyon fonksiyoneli sunulmakta ve hareketli bir sensörün, çevresi hakkında topladığı toplam bilgiyi iyileştirmek amacıyla sonraki konumunu tahmin ettiği bir sensör yönetimi problemi için kullanılmaktadır.

Anahtar Kelimeler: Dağıtık Füzyon, Sensör Yönetimi, Durum Kestirimi, Rastgele Sonlu Kümeler.

ACKNOWLEDGEMENT

I want to express my gratitude to Karamel and Satoshi for their unconditional love and support. They kept me motivated and were always right there during the writing of this dissertation.

TABLE OF CONTENTS

	<u>Page</u>
SUMMARY	vi
ÖZET	vii
ACKNOWLEDGMENT	viii
TABLE OF CONTENTS	ix
ABBREVIATIONS	xiv
LIST OF FIGURES	xvi
LIST OF TABLES	xviii
1. INTRODUCTION	1
1.1. Motivation and Scope	5
1.2. Contributions	7
1.3. Thesis Outline	8
2. FUNDAMENTALS: BAYESIAN DATA FUSION, NAVIGATION, AND TRACKING	11
2.1. Recursive Bayesian Filtering	11
2.1.1. General Non-linear Bayes Filter	15
2.1.2. Moment Approximations of the Bayes Filter	16
2.1.2.1. Kalman Filter	16
2.1.2.2. Extended Kalman Filter	20
2.1.2.3. Unscented Kalman Filter	21
2.1.3. Density Approximation of the Bayes Filter	22
2.1.3.1. Particle Filter	23
2.2. Multi-Target Tracking	25
2.2.1. Measurement-to-Track Association (M2TA)	27
2.2.2. Joint Probabilistic Data Association (JPDA)	28
2.2.3. Multiple Hypothesis Tracking (MHT)	29
2.3. Random Finite Sets (RFS)	31
2.3.1. FISST Multi-Target Calculus	31
2.3.1.1. Set integral	32

2.3.1.2. Probability generating functionals	32
2.3.1.3. Functional derivative	33
2.3.2. Random Finite Set-based Filters	34
2.3.2.1. Probability Hypothesis Density (PHD) Filter	36
2.3.2.2. Generalized Labelled Multi-Bernoulli Filter	38
2.3.2.3. Poisson Multi-Bernoulli Mixture Filter	40
2.4. Multi-Sensor Fusion	43
2.4.1. Centralized Fusion	44
2.4.2. Distributed Fusion	46
2.4.2.1. Exponential Mixture Fusion (Geometric Average)	47
2.4.2.2. Arithmetic Average Fusion	48
2.5. Performance Metrics	49
2.5.1. Multi-Object Miss Distance	49
2.5.1.1. Hausdorff Distance	50
2.5.1.2. Wasserstein Distance	50
2.5.1.3. Optimal Subpattern Assignment (OSPA) metric	51
2.5.2. Multi-Object Information Functionals	52
2.6. Integrated Navigation	53
2.6.1. Integrated (Strapdown) Navigation Equations	54
2.6.1.1. Velocity Equation	55
2.6.1.2. Attitude Equations	56
2.6.1.3. Position Equations	57
2.6.2. Inertial Navigation System Error Model	57
2.6.2.1. Velocity Error Equations	57
2.6.2.2. Attitude Error Equations	58
2.6.2.3. Position Error Equations	59
2.7. Appendices	60
2.7.1. Multi-Object Probability Distributions	60
2.7.2. Important Multi-Object RFSs	61
2.7.2.1. Independent, Identically Distributed (i.i.d.) Cluster Process	62
2.7.2.2. Multi-Object Poisson Process	62
2.7.2.3. Multi-Object Dirac Delta Density	62
2.7.2.4. Multi-Object Uniform Process	63

2.7.2.5. Multi-Bernoulli Process	63
3. LEVEL 2 FUSION: MEASUREMENT-TO-TRACK FUSION	65
3.1. Literature Overview	66
3.2. Standard Measurement Model: Point-Target	70
3.3. Generalized Data Modeling	71
3.3.1. Generalized Likelihood Functions	73
3.4. Cluster-Target Tracking	75
3.4.1. Markov Transition Density for Cluster-Target Model	76
3.4.2. Bayes Update for the Cluster-Target Model	81
3.5. Bayesian Nonparametric Modeling: Dirichlet Processes	85
3.6. Numerical Studies	86
3.6.1. Synthetic Scenario-1	87
3.6.2. Synthetic Scenario-2	89
3.6.3. Real flight scenario	94
4. LEVEL 4 FUSION: MULTI-TARGET SENSOR MANAGEMENT	101
4.1. Introduction	101
4.2. Literature Review	102
4.2.1. Partially Observed Markov Decision Process (POMDP)	104
4.2.2. Objective Functions	105
4.2.2.1. Task-Driven Approach	106
4.2.2.1.1. Posterior Expected Number of Targets (PENT)	106
4.2.2.1.2. Cardinality Variance	107
4.2.2.1.3. Posterior Expected Error of Cardinality and States (PEECS)	108
4.2.2.2. Information-Theoretic Approach	108
4.2.2.2.1. Shannon Entropy	109
4.2.2.2.2. Kullback-Leibler Divergence	109
4.2.2.2.3. Rènyi Divergence	110
4.2.2.2.4. Cauchy-Schwarz Divergence	110
4.2.2.2.5. Hölder Divergence	111
4.3. Extended Hölder Divergence (EHD) for PPP	112
4.3.1. Implementation of EHD for GM Models	115
4.3.2. Sequential Monte Carlo Implementation of EHD	117
4.4. Application of EHD to Multi-Target Sensor Management	118

4.4.1. Numerical Studies	120
4.4.1.1. Scenario 1	122
4.4.1.2. Scenario 2	124
4.4.1.3. Scenario 3	128
4.5. Conclusions	132
4.6. Appendices	132
4.6.1. Derivation of (4.31)-(4.33)	132
4.6.2. Derivation of (4.38)-(4.40)	134
5. LEVEL 5 FUSION: SENSOR FUSION FOR ASSURED NAVIGATION	136
5.1. Aided Navigation	137
5.1.1. Literature Summary	137
5.2. Terrain and Visual Aiding Applications	141
5.2.1. Definition of the Dynamical Model	141
5.2.2. Visual Aided Navigation	142
5.2.2.1. Problem Definition	143
5.2.2.2. Camera Measurement Model	144
5.2.2.3. Visual Aiding Application to Inertial Navigation	146
5.2.3. Terrain Aided Navigation	148
5.2.3.1. Problem Definition	148
5.2.3.2. DTED Maps	150
5.2.3.3. Radar Altimeter Measurement Model	150
5.3. Multi-Sensor Fusion for Assured Navigation	151
5.4. Numerical Studies	153
5.4.1. Scenario-1	154
5.4.2. Scenario-2	158
5.4.3. Scenario-3	162
5.5. Conclusions	165
6. CONCLUSIONS AND FUTURE WORKS	166
6.1. Conclusions	166
6.2. Future Works	168
REFERENCES	170
BIOGRAPHY	186

ABBREVIATIONS

<u>Abbreviations and Acronyms</u>	<u>Explanations</u>
AA	: Arithmetic Average
3D	: 3-Dimensional
CI	: Covariance Intersection
CSD	: Cauchy-Schwarz Divergence
CPHD	: Cardinalized Probability Hypothesis Density
CRP	: Chinese Restaurant Process
CT-PHD	: Cluster-Target Probability Hypothesis Density
DDF	: Distributed Data Fusion
DEM	: Digital Elevation Model
DP	: Dirichlet Process
DPMM	: Dirichlet Process Mixture Model
DTED	: Digital Terrain Elevation Data
EAP	: Expected a Posteriori
ECEF	: Earth-centered, Earth-fixed
EHD	: Extended Hölder Divergence
EKF	: Extended Kalman Filter
FISST	: Finite Set Statistics
GA	: Geometric Average
GLF	: Generalized Likelihood Function
GLMB	: Generalized Labeled Multi-Bernoulli
GM	: Gaussian Mixture
GNSS	: Global Navigation Satellite System
GPS	: Global Positioning System
HO-MHT	: Hypothesis Oriented MHT
IMM	: Interacting Multiple Model
IMU	: Inertial Measurement Unit
INS	: Inertial Navigation System
JDL	: Joint Directors of Laboratories
JPDA	: Joint Probabilistic Data Association

KLD	:	Kullback-Leibler Divergence
M2M	:	Measurement-to-Measurement
M2T	:	Measurement-to-Track
MAP	:	Maximum a Posteriori
MHT	:	Multiple Hypothesis Tracking
MOT	:	Multi Object Tracking
MSL	:	Mean Sea Level
MS-MT	:	Multi-Sensor Multi-Target
MTT	:	Multi-Target Tracking
NLS	:	Nonlinear Least Squares
OSPA	:	Optimum Subpattern Assignment
TDOA	:	Time Difference of Arrival
UKF	:	Unscented Kalman Filter
PDA	:	Probabilistic Data Association
PENT	:	Posterior Expected Number of Targets
PEECS	:	Posterior Expected Error of Cardinality and States
p.g.fl.	:	Probability Generating Functional
PHD	:	Probability Hypothesis Density
PIMS	:	Predicted Ideal Measurement Set
POMDP	:	Partially Observed Markov Decision Process
PPP	:	Poisson Point Process
RFS	:	Random Finite Sets
SLAM	:	Simultaneous Localization and Mapping
SMC	:	Sequential Monte Carlo
SRTM	:	Shuttle Radar Topography Mission
T2T	:	Track-to-Track
TAN	:	Terrain Aided Navigation
TO-MHT	:	Track Oriented MHT
UAV	:	Unmanned Aerial Vehicle
UKF	:	Unscented Kalman Filter
UT	:	Unscented Transform
VAN	:	Visual Aided Navigation

LIST OF FIGURES

<u>Figure No:</u>	<u>Page</u>
2.1: Conventional MTT architecture.	26
2.2: Two MHT variations in implementation.	30
2.3: An example of data association uncertainty with two tracks and three observations.	30
2.4: Complete data-fusion model and relation of each level with each other.	43
2.5: Centralized fusion architecture: measurement fusion.	44
2.6: Distributed fusion architecture: state fusion.	46
2.7: A generic integrated navigation architecture.	54
2.8: Strapdown navigation mechanization.	55
3.1: Bayesian processing of generalized measurement data.	73
3.2: Ground-truth (synthetic) target trajectories.	87
3.3: CT-PHD filter cardinality estimates.	89
3.4: Measurements and estimates in each axis.	90
3.5: Estimates vs. ground truths.	90
3.6: Performance metric OSPA cardinality, localization, and total errors for scenario-1.	91
3.7: Ground truth trajectories for synthetic Scenario-2.	92
3.8: Estimated target trajectories and measurements w.r.t. each axes.	93
3.9: Estimated cardinality vs. ground truth cardinality for Scenario-2.	94
3.10: Estimated target trajectories for Scenario-2.	95
3.11: Performance metric OSPA total, localization, and cardinality errors.	96
3.12: Measurement clusters at $k = 45$ s.	97
3.13: Ground-truth target trajectories and receiver positions.	97
3.14: Measurement clouds and partitioned cluster centers.	98
3.15: Histogram plot of the number of detections for each target in a single scan.	99
3.16: Tracking result of the 10-target scenario using the proposed algorithm.	99
3.17: Estimated number of targets in the scenario.	100
4.1: Taxonomy of sensor management/control approaches for MTT.	104
4.2: Markov models and their relation to state estimation.	105

4.3: Illustration of moving sensor control concept for multi-target tracking.	119
4.4: Flow chart of the sensor management scheme.	120
4.5: Total OSPA error for the three reward functions and the stationary sensor case.	123
4.6: Average probability of detection profile for each reward function.	124
4.7: Total OSPA error, an average of 100 MC runs for EHD $\beta = \{1.2, 10.2\}$ and Rényi $\alpha = \{0.15, 0.5, 0.9\}$ for a dynamic environment.	125
4.8: Total OSPA error, an average of 100 MC runs for EHD $\beta = \{1.2, 10.2\}$ and Rényi $\alpha = \{0.15, 0.5, 0.9\}$ for the static environment.	126
4.9: Sensor locations for different reward functions.	127
4.10: Effect of γ parameter selection on the sensor control performance, $\beta = 1.2$.	128
4.11: Scenario-2 ground truth trajectories. The targets are born inside the particle clouds and die at the red triangles.	129
4.12: Cardinality estimation result of different information-theoretic reward functions.	130
4.13: OSPA Cardinality Error Comparison.	131
4.14: OSPA Localization Error Comparison for $\beta = 1.2$.	131
5.1: Camera-navigation space geometry.	145
5.2: VAN algorithm block diagram.	147
5.3: Radar altimeter measurement in TAN.	149
5.4: Multi-sensor distributed fusion architecture for assured navigation.	152
5.5: Scenario-1 ground truth (red) and estimated trajectory (blue).	155
5.6: Scenario-1 error bounds and actual error.	156
5.7: VAN (top row), TAN (middle row), and GA fusion (bottom row) 3σ bounds.	157
5.8: Scenario-1 free inertial error and fused estimation error.	158
5.9: Scenario-2 ground truth (red) and estimated trajectory (blue).	159
5.10: Scenario-2 error bounds.	160
5.11: VAN (top row), TAN (middle row), and GA fusion (bottom row) 3σ bounds.	161
5.12: Scenario-2 free inertial error and fused estimation error.	161
5.13: Scenario-3 ground truth (red) and estimated trajectory (blue).	163
5.14: Scenario-3 error bounds.	164
5.15: Scenario-3 VAN (top row), TAN (middle row), and GA fusion (bottom row) 3σ bounds.	164
5.16: Scenario-3 free inertial error and fused estimation error.	165

LIST OF TABLES

<u>Table No:</u>	<u>Page</u>
4.1: Comparison of average OSPA errors for three reward functions.	132
5.1: Sensor fusion scenario parameters.	154

1. INTRODUCTION

In contemporary society, an array of sensing devices permeates our daily existence, offering convenience and enhancing various aspects of our lives. These sensors monitor our everyday activities, ensure residential security against multifarious threats such as burglary and fire, facilitate the management and operation of autonomous vehicles, and contribute to defense and public security applications through technologies such as radars and video cameras. Numerous applications employ diverse sensor types, each generating distinct data sets, to efficiently accomplish their objectives. Consequently, the system must process the voluminous data emanating from these varied sensors, necessitating substantial processing power, memory, and speed. This circumstance engenders the imperative to minimize data volume, harmonize disparate data types, and amalgamate them into a unified entity upon which the machine (or system) can base its decision. Thus, the data fusion problem emerges as a critical engineering discipline.

Data fusion can be categorized into several different classification schemes or models that can be found in the review paper by Castanedo [1] Classification can be done regarding the relations between data provided by sources as complementary, redundant, or cooperative [2]. It can be based on the input/output data types as proposed in [3], an abstraction level of the data [4], or the architecture type. Amongst these models, the most used classification model is the one that is proposed by Joint Directors of Laboratories (JDL) [5]. This model classifies the data fusion into five levels [1]:

- Level 0 – Data fusion at the raw signal and pixel level, known as source preprocessing, reduces the amount of data and preserves useful information for processing.
- Level 1 – Object refinement involves various techniques, such as spatial and temporal alignment, association, correlation, and state estimation, which are commonly used in object tracking algorithms.
- Level 2 – Situation assessment, which involves making inferences at a higher level, involves determining the relationships between objects in order to identify patterns.
- Level 3 – Impact assessment involves evaluating the impact of detected activities

from lower levels and identifying operational opportunities, risks, and future projections.

- Level 4 – Process refinement involves optimizing resource and sensor management, sensor control, and scheduling activities in order to improve the performance of previous levels.
- Level 5 – Cognitive refinement involves optimizing the interactions between data fusion systems and human operators [6].

Data fusion is incorporated into a diverse set of applications, including robotics [7], defense [8], aerospace, bioinformatics [9], etc. In military applications, the (multi-sensor) data fusion or information fusion is also known as the process of jointly detecting, tracking, and identifying multiple targets using multiple sensors [10]. Multi-Sensor Multi-Target (MS-MT) tracking, a type of data fusion, involves combining information from multiple sources to produce more accurate and useful information than any individual sensor could provide.

Data fusion algorithms used in multi-sensor multi-target tracking systems are classified as Level 1 in the JDL model. These algorithms can be further divided into three categories based on the tracking step in which they are used: measurement-to-measurement fusion, measurement-to-track fusion, and track-to-track fusion.

- Measurement-to-measurement (M2M) fusion involves combining measurements from multiple sensors into a single, global measurement that can be used in a state estimator to update or create a track [11]–[13].
- Measurement-to-track (M2T) fusion combines measurements with existing tracks that they are associated with, which is what a tracking filter, such as a Kalman filter, does.
- Track-to-track (T2T) fusion combines information from individual sensors at the output of tracking filters, with each sensor using its own tracking filter and sharing track-level information with other sensors.

Track fusion algorithms can be classified into three main categories based on their system architecture: centralized, decentralized, and distributed. In a centralized architecture, a central node, known as the fusion node, collects information from other nodes at the track level and applies the track fusion algorithm. This process can also be done at the observation level under certain assumptions such as correct data alignment

and association. However, centralized architecture can be vulnerable to single points of failure and may be impacted by time delays in data transfer. In a decentralized architecture, data fusion is performed at each individual node, which has its own processing capabilities and communicates with a central node at the track level [14]. This architecture requires more computation and communication resources but does not have a single point of failure. A distributed architecture is the most suitable for track-to-track fusion and involves multiple nodes working together to perform data fusion.

Despite being separated by Castanedo in the review paper [1], decentralized and distributed architectures are very similar in process to each other and some studies can be found in the literature that the terms decentralized and distributed are used to describe the same process [15]–[18]. Also, in [19], [20] the architectures are separated as centralized and distributed. In this work, distributed track fusion term is adopted for the aforementioned process.

Before the fusion process, in a distributed network, target tracks must be estimated at each node. In an MS-MT tracking scenario, there are some challenges that must be handled prior to fusion. These challenges are detection uncertainty, clutter, and association uncertainty [21]–[24]. A multi-object tracking (MOT) algorithm should account for these problems. From this point of view, the multi-object tracking algorithms proposed in the literature can be in three major concepts: Joint Probabilistic Data Association (JPDA) [25]–[27], Multiple Hypothesis Tracking (MHT) [28] and Random Finite Sets (RFS) [21], [29]–[32].

In addition, automated sensor management which is described by Mahler [33] as “the process of directing the right sensors to the right targets at the right times” is an important step in MS-MT systems. This process can be categorized into the “Level 4 data fusion” process. A sensor management system must simultaneously take into account a number of factors that affect performance. These factors include the current and anticipated future characteristics of targets, network constraints, and even the evolution of operators’ priorities and tactical goals [33]. This means that resource management requires the integration of several functionalities, such as multi-target detection, localization, tracking, identification, MS-MT information fusion, etc.

Most of the conventional MS-MT tracking systems assume a point-target detection for the sensor model. This model assumes that each target generates a single detection and each detection is generated by a single target. However, in practical applications, this assumption may not hold. For example, in visual target

tracking pixelized targets are usually represented with multiple pixels as detections, which is referred to as the extended target [34]–[36]. For this particular problem, the standard point-detection model is not valid. As another example, low-resolution radars and sonars may not be able to discriminate closely spaced targets and therefore fails to produce separate detections, thus causing degradation in the overall system performance. This situation is referred to as the merged measurements. For these, and many other, reasons employing non-traditional sensor models in MS-MT tracking systems may improve the system performance, especially when the human factor is involved in obtaining the observations [33].

Another critical application of the data fusion process is the inertial navigation problem. In navigation, inertial sensors provide the main kinematic information of the platform. The inertial information from the accelerometer and gyroscope readings are numerically integrated into position, velocity, and attitude information [37]. Numerical integration, however, results in the errors accumulating over time, and in the end to diverge from the true position. Therefore, it is extremely necessary to provide aiding information from other sensor sources to limit and correct inertial errors. Thus, data fusion is an integral part of navigation problems. Since the data being fused are usually processed in individual sensor modalities, such as satellite positioning, visual positioning, radio navigation, etc. the fusion process in this problem can be categorized into Level 5 of JDL. In other words, the estimates (or posterior distributions) from each sensor are used in the fusion process.

Data fusion problems usually encompass dynamic state estimation problems which are almost always needed to run online. Because the real-time, sequential processing requirements, model-based Bayesian techniques have been studied extensively in the data fusion field. However, with the development of processing technology and proliferation of sophisticated deep learning frameworks, application of data-driven approaches such as [38]–[41], are gaining attention. Although there are studies that report superior performance of the data-driven approaches against model-based techniques, there are problems that are still needed to be addressed such as huge data and labeling need in the training phase, extensive processing power requirements of the large networks, etc. Therefore, the model-based Bayesian methods still dominate the dynamic estimation literature to this date.

1.1. Motivation and Scope

The impetus for this research is derived from the necessity to scrutinize the Bayesian data fusion methodology from a more elevated perspective and explore its various facets. Owing to the continuous advancements in sensor technology, it is feasible to integrate a multitude of sensor modalities across diverse applications. As individual sensors are inadequate in delivering information of satisfactory quality, the employment of multiple sensor data is indispensable for sensor data processing. Given that this integration can transpire across an array of stages, from raw signal to system level as delineated by the Joint Directors of Laboratories (JDL) model, the fusion process warrants suitable implementation. This dissertation addresses the data fusion quandary in a tripartite structure, encompassing background, algorithms, and applications as distinct components. While data fusion systems hold potential for numerous fields of application, this dissertation specifically concentrates on surveillance applications, with an emphasis on navigation and object tracking.

The concept of Bayesian data fusion is rooted in the Bayesian probability theory, which provides a comprehensive framework for integrating information from various sources in a principled manner. In the context of data fusion, the Bayesian approach enables the combination of diverse sensor data to enhance the overall quality and reliability of the obtained information.

A primary advantage of the Bayesian data fusion methodology lies in its ability to accommodate uncertainties and inconsistencies inherent in the data from disparate sensor sources. By leveraging the Bayesian inference mechanism, the method estimates the posterior probability distribution of the latent variables, taking into account both the prior distribution and the likelihood of the observed data.

The first part of this dissertation is dedicated to Level 1/2 fusion with an application of measurement-to-track fusion. Different sensor modalities usually have different characteristics and limitations. Due to various reasons such as data collection costs or confidentiality, the specific data forms of the surveillance systems cannot be exposed. Also, from the computational viewpoint, the processing capabilities have been insufficient until recently, thus computationally demanding applications have been mostly overlooked. For these reasons, most of the studies in the target tracking literature focus on the standard small target model where it is assumed that a single target generates a single measurement at any time, as well as a single measurement can only be originated from a single target [32],[42]–[44]. However, this so-called standard

model only encompasses a relatively small number of applications. The extended targets [45], [46], group targets [47], [48], cluster targets [35], [49], [50], quantized measurements [51], [52] are some studies where non-traditional measurements are investigated. As can be seen in the above references, non-traditional measurements which are the types of measurements that fall outside of the standard model definition, are very common to encounter in practical systems. We dedicate this chapter to the cluster-target tracking problem where the targets generate multiple measurements in each scan with no particular shape. The problem is particularly important in passive tracking and applications where point-cloud measurements are used.

The second part investigates the data fusion Level 3/4 which is applied to improve the performance of the system. The quality of sensor observations depends not only on the environment and the objects that the sensor monitors, but also on the state of the sensor (and/or the platform) itself. By knowing how its own state can affect the observation process or by directing its limited resources to a more tactically important objective, a sensor can provide a more reliable view of the surveillance area. This process is widely known as sensor management/control [53]–[57]. The most common application, in object tracking, of the sensor management process is where the sensing modality is mounted on a moving platform. The state or position of the sensor platform ultimately affects the quality of measurements it received, even more so when there are multiple targets in the surveillance area to be tracked. In this context, some targets may be more tactically important. We may want to remain in the vicinity of some targets to better localize them or to prevent detection and maintain stealthiness, not to come closer to some. Also, sensor management can be applied to navigation problems in such a way that the platform can be better localized. These kinds of constraints on the sensor platform can be realized by sensor management algorithms. By designing appropriate sensor management systems, the sensor resources can be directed to maximize some specific objective(s).

The third and final part is focused on Level 5 fusion which involves the combining posterior information of different sensor processing units. This process aims to provide a more robust view of the environment and estimated quantities by fusing statistical outputs of different sensor modalities. The process is called posterior or estimation fusion [58]–[62]. Because the fusion process at this level involves outputs of several sensor processing units/nodes, the output data format of different sensor nodes are mostly the same, i.e. in a posterior probability distribution form or its moments such as mean and covariance information. Thus, unlike low-level

information fusion, which mostly focuses on trying to map the different sensor information to a common space as well as reducing uncertainty, the purpose of posterior fusion is to increase the robustness and integrity of the fused solution with respect to individual sensor outputs. This fusion level is especially useful in navigation problems where two or more sensor information are fused into a single estimation which is more robust and reliable.

1.2. Contributions

The first contribution of the dissertation is the application of a novel information-theoretic divergence to the multi-target sensor control problem in the random finite set framework. First, the analytical form of the Hölder divergence is derived for Poisson point processes and applied to the multi-target tracking problem with a moving sensor platform for observer trajectory optimization in a sequential Monte Carlo implementation. Then, for the case where intensity functions of the Poisson point processes have Gaussian mixture forms, the analytical closed-form expressions of the proposed (extended) Hölder divergence are derived. Hölder divergence generalizes the well-known Cauchy-Schwarz divergence, and by simulations, it is shown that the proposed Hölder divergence-based sensor control algorithm is more flexible in certain scenarios. Preliminary results indicate that the proposed algorithm can be adapted to changing situations, by parameter tuning, such as static or dynamic-target environments.

The second contribution is the novel application of the posterior fusion of different sensor modalities, for navigation aiding, in a distributed/decentralized fusion concept. The sensors that are being fused have different characteristics. The heterogeneous data includes the visual and terrain-aiding sensors. The heterogeneous sensor data are fused together to obtain more robust position information of the aircraft. Since both the aiding schemes use the inertial error model, common prior information obtained from the inertial navigation system, so-called double counting occurs and must be handled properly by the fusion algorithm. Direct averaging or a Kalman-based information fusion method suffers from the data-incest problem. Borrowed from the target-tracking literature, Chernoff fusion and arithmetic average fusion algorithms, bypassing the so-called double counting, are applied to the aided navigation problem in a distributed manner. It provides a robust solution by fusing complementary

sensor information and increases the robustness of the navigation system to outside interferences. To the best of our knowledge, there is no study in the navigation field regarding this problem. Therefore, it would be a valuable contribution to the field.

The third contribution encompasses a case study focusing on low-level data fusion, specifically addressing the problem of cluster-target tracking, representative of non-traditional sensor models. In situations where targets within a surveillance zone yield multiple measurements per target, typically resulting from an ambiguous measurement generation process, the employment of standard measurement models becomes unfeasible. Mathematically, these measurement clusters give rise to generalized measurement models. By employing the principles of generalized measurements and the random finite set framework, a cluster-target tracking algorithm is formulated, which enables the effective fusion of cluster measurements with target tracks. The mathematical expressions for the dynamic and measurement models are derived, encapsulating the evolution of cluster shape, size parameters, and the cluster center. A principal challenge lies in estimating clusters when the number of clusters remains uncertain. This issue is addressed through the utilization of nonparametric Bayesian estimation techniques, specifically employing the Dirichlet process mixture for the clustering process. Subsequently, the devised algorithm is implemented using real-world sensor data, and its performance is evaluated accordingly.

A supplementary contribution of this research is the asynchronized association of different sensor data which is a crucial step for sensor posterior fusion and is usually overlooked. It is plausible that different sensor modalities process their data at different rates. In this case, their respective outputs are not synchronized. Thus, it is important to synchronize these outputs to a common time before the fusion process.

1.3. Thesis Outline

The outline of the thesis is as follows:

In Chapter 2, an in-depth exploration of the Bayesian data fusion background is provided. A taxonomic representation of fusion concepts utilized in Bayesian estimation problems is presented. Subsequently, the chapter delves into various levels of the general fusion problem, including measurement fusion (estimation) and estimation fusion concepts. In the context of measurement fusion, Bayesian single and multi-target tracking problems and their corresponding methods are

examined thoroughly. State-of-the-art techniques are summarized for both single and multi-target tracking issues. As this dissertation embraces the random finite set framework for modeling, multi-target calculus elements are introduced in this chapter. Following this, estimation fusion (or posterior fusion) concepts are explicated in detail. Diverse fusion architectures, such as centralized and decentralized fusion, are deliberated upon, with a discussion of their respective merits and drawbacks. The estimation fusion is investigated through the lens of the inertial navigation problem. Fundamentally, navigation and target tracking are closely related issues in which sensor measurements are employed to monitor the kinematics of the ownship/platform for navigational purposes. From a data fusion perspective, these two problems are identical. Consequently, the concluding section of the chapter presents the basics of inertial navigation in the context of multi-sensor fusion for assured navigation (i.e., GNSS-denied environments). The objective of this chapter is to offer a comprehensive mathematical foundation for Bayesian data fusion.

In Chapter 3, measurement fusion is studied and demonstrated through the multi-target tracking problem for non-traditional measurement models. A review of the approaches found in the literature is summarized. The definition of the non-traditional measurements is presented which are all the measurement models that do not conform to the standard measurement model where *each target generates only one measurement and each measurement is originated from a single target*. It is also referred to as *the generalized measurements*. We propose a multi-target tracking algorithm for non-traditional measurements, specifically using cluster measurements (also known as dynamic clustering) in which each target generates multiple measurements which can have an arbitrary scattering shape. The proposed cluster target tracking algorithm is based on the random finite set framework which can naturally accommodate the generalized measurements in multi-target tracking. Dynamic clustering is done via Dirichlet process mixture which is a Bayesian nonparametric model. It provides the necessary tool to cluster and associate the measurement set into an unknown number of target clusters. Multi-target tracking simulations conducted in the presence of cluster targets are presented, based on both simulated and real target measurements.

Chapter 4 focuses on the multi-target sensor management problem, i.e. Level-4 fusion. Information-theoretic and mission-oriented management approaches are discussed and different methods proposed in the literature are summarized. A novel information-theoretic sensor management method based on Hölder divergence is proposed for the Poisson point processes, called extended Hölder divergence (EHD).

The proposed method is applied to moving sensor control problems in a multi-target environment. Via simulations, the proposed method is compared against similar methods found in literature and is shown to be superior.

Chapter 5 discusses the multi-sensor fusion for assured navigation, which is a critical topic for self-localization, using onboard sensors when satellite signals are not available, as a level 5 fusion concept. A literature review on navigation-aiding schemes is summarized in this chapter. A novel sensor fusion application has been presented that utilizes inertial sensors, a radar altimeter, and visual sensors. Borrowing from the multi-sensor target tracking discipline, posterior fusion methods are applied in aiding the inertial sensor data, in a distributed manner. By simulation, it is shown that the application of the posterior fusion to the aided navigation improves the navigation performance and redundancy.

Finally, in Chapter 6, conclusions are drawn from the conducted studies. Some potential future research topics are proposed to further advance the research discussed in this dissertation.

2. FUNDAMENTALS: BAYESIAN DATA FUSION, NAVIGATION, AND TRACKING

In this chapter, we provide the necessary background material which is needed to properly assess the remainder of this dissertation. We focus on different aspects of the general problem of "data fusion" which was summarized in the previous chapter. The chapter presents general background material of the three main topics, each representing the data fusion at a different level as described in [5]. First is the recursive Bayesian filtering, which can be considered as the building block of Bayesian data fusion. We present the general (nonlinear) Bayesian filtering equations from which single and multi-target filters are derived. The most widely used estimation filters for single and multi-target problems are discussed next. Thus, section 2.1. focuses on Level 1 data fusion which consists of state estimation problems, in general. In section 2.4., the sensor fusion problem is described and a literature review is presented. The section is addressed in two parts: centralized and distributed fusion, regarding how the fusion process is carried out. Last, in the chapter, inertial navigation is briefly summarized since a sensor fusion application is proposed for aided navigation later in the dissertation.

2.1. Recursive Bayesian Filtering

It may be considered that the history of information fusion has begun with the publication of Bayes' theorem in 1763 [63] and Gauss' least squares method in 1795 (or in 1809) to estimate the orbits of celestial bodies, using observations, instead of Kepler's equations of planetary motion. Since then, extensive research has been done for information fusion and applied to various disciplines including signal and image processing, pattern recognition, etc. An important step in the history of information fusion is the development of the Kalman filter in 1960 [64] for sequential estimation of object dynamics.

Target tracking is the process of estimating the state of a target over time. The state of a target can include its position, velocity, and other attributes. Target tracking is a challenging problem because the target's state can change over time due to factors such as movement, occlusion, and noise.

There are many different algorithms for target tracking. One of the most common algorithms is the Kalman filter. The Kalman filter is a recursive Bayesian filter that uses a linear model of the target's state to estimate the target's state. The Kalman filter is effective for tracking targets that move in a linear fashion. However, the Kalman filter is not effective for tracking targets that move in a nonlinear fashion.

The choice of which algorithm to use for target tracking depends on the specific application. For applications where the target is expected to move in a linear fashion, the Kalman filter is a good choice. For applications where the target is expected to move in a nonlinear fashion, the EKF is a good choice.

For target tracking, many algorithms that incorporate the new observations about the target with the existing information have been proposed in the literature. Kalman filter is used for online estimation of target states for linear-Gaussian problems and its information matrix filter (or inverse covariance filter) forms [65] for track-to-track fusion. The Kalman filter is the optimal estimate for the linear system models and with additive, independent white noise in both dynamic and measurement models. When the problem is not linear, which is a more realistic case for real-life applications and one of the essential problems in target tracking, the standard Kalman filter is ineffective. In this case, by applying a linearization scheme to the non-linear functions in the model, a locally linear version of the Kalman filter called the extended Kalman filter (EKF) is proposed [66], [67]. The EKF is an analytic approximation method for non-linear filtering. The non-linear functions are approximated by the first term in their Taylor series expansion. If the non-linear functions are discontinuous or non-linearity is severe, the performance of the EKF will significantly reduce.

In addition to the Kalman filter and EKF, there are a number of other methods for solving non-linear tracking problems. These methods include numerical approximations, Gaussian sum or multiple model filters, and sampling approaches.

Numerical methods approximate the solution to the non-linear problem by discretizing the integration variable [68]. Gaussian sum filters [69]–[71] construct the posterior density using a mixture of Gaussians. The interacting multiple models (IMM) algorithm [72], [73] uses multiple dynamic models to track a target. The unscented Kalman filter (UKF) and its variants use a set of deterministically chosen points to approximate the posterior and propagate through non-linear functions [74]–[76]. Particle filters and their variants represent the posterior using point masses (particles) and associated weights and perform sequential Monte Carlo (SMC) estimation on these points [18], [71], [77]–[79].

These methods are all effective for solving non-linear tracking problems, and the choice of which method to use depends on the specific application. For example, numerical methods are often used for problems where the non-linearities are mild, while Gaussian sum filters are often used for problems where the non-linearities are severe. The IMM algorithm is often used for maneuvering target tracking, while the UKF and particle filters are often used for bearings-only target tracking. Multi-target tracking (MTT) is a challenging problem due to the following factors:

- Detection uncertainty: The uncertainty of whether a target generated an observation or not, due to factors such as occlusion, low radar cross-section, or low signal-to-noise ratio.
- False alarms: Observations that are not generated by any target, but are caused by factors such as multipath, other objects in the region, or clutter.
- Data association uncertainty: The difficulty of determining which target generated which measurement.

To address these challenges, multi-target tracking algorithms in literature can be grouped into three main categories:

- Joint Probabilistic Data Association (JPDA): An algorithm that uses a joint probability model to associate measurements with targets.
- Multiple Hypothesis Tracking (MHT): An algorithm that generates multiple hypotheses about the association of measurements with targets and then tracks each hypothesis separately.
- Random Finite Set (RFS) approach: An algorithm that uses a generative model to represent the set of possible target states and measurements.

In this literature summary, multi-target tracking methods under these paradigms will be presented.

Data association uncertainty is a challenge in multi-target tracking that arises when the origin of observations is uncertain [26]. This can occur when detection quality is low and clutter and spurious measurements are also detected, or when multiple targets are in close proximity to each other. Using invalid measurements to update existing tracks can lead to an increase in estimation error.

To mitigate this problem, it is important to identify which measurements should be used to update the tracks. The Joint Probabilistic Data Association (JPDA) filter is a commonly used algorithm for data association [26]. The JPDA filter calculates

association probabilities for each validated measurement to the target being tracked. Validated measurements are the measurements that are in close proximity to the predicted target state. The process of identifying validated measurements for a track is called measurement validation or gating.

The JPDA filter can be based on a Kalman filter or an extended Kalman filter, depending on whether the problem is linear or non-linear. Various forms of JPDA filters have been proposed in the literature, including those that incorporate fuzzy logic [80], incorporate track quality criteria to improve association, or can handle multiple detections from the same target falling into the validation region [81].

The multiple hypothesis tracking (MHT) [28] method is a multi-target tracking algorithm that forms multiple hypotheses for the measurement-to-track association, including conflicting ones. These hypotheses are propagated over time with the expectation that future observations will resolve the uncertainty of the associations [82]. There are two main variants of the MHT algorithm: the hypothesis-oriented (HO-MHT) method [28] and the track-oriented (TO-MHT) method [83], [84]. The main difference between the two methods is that the hypothesis-oriented method propagates hypotheses over time, while the track-oriented method expands the existing track set using all possible associations. The MHT approach inherently handles the process of the track initiation and termination [85]. However, since the number of hypotheses can grow exponentially over time, additional operations like hypothesis pruning must be incorporated into the algorithm to maintain computational traceability [16].

Mahler first introduced a method known as the random finite set (RFS) approach for multi-target tracking (MTT) [21]. This technique uses mathematical tools called finite set statistics (FISST) [32], [86]–[89] to tackle the MTT challenge. Vo's dissertation [21] provides an examination of the RFS approach using point process theory. This method, although relatively recent in the MTT field, has garnered substantial interest. Unlike the other two methods previously discussed, the RFS approach eliminates the need for data association. It is essentially an expanded version of the standard Bayes filter for multi-target problems. Its aim is to estimate the variable count and states of objects over time [23]. According to Mahler in his book "Statistics 102", RFS algorithms can achieve genuine tracking. This implies that a precise, closed-form and computationally efficient solution to the MTT problem is available and can be proven to be optimal Bayes [89]. In the RFS technique, multi-target states are depicted as a finite set, a representation considered fitting for the MTT problem.

The Random Finite Set (RFS) approach to Multi-target Tracking (MTT) is mainly based on two different methods: moment and cardinality approximations, and density approximations. The moment approximation methods include the Probability Hypothesis Density (PHD) filters [34], [70], [90], [91] and its cardinalized version, the Cardinalized PHD (CPHD) filters [58], [92], [93], that use Gaussian mixture and Sequential Monte Carlo (SMC) implementations. Density approximations are based on Multi-Bernoulli (MeMBer) filters [30], [94], labeled and generalized labeled MeMBer (GLMB) filters [95]–[97], with their Gaussian mixture and SMC implementations. However, a limitation of most RFS approaches is that they only provide point estimates of targets that are not connected in time. The labeled RFS filters overcome this limitation by augmenting the state vector with a label specific for each track, and thus they estimate the label as well as the track state in time. The GLMB filter is considered a provably Bayes-optimal and computationally tractable multi-target tracking filter [32].

2.1.1. General Non-linear Bayes Filter

The recursive Bayes filter is the primary tool for many engineering applications such as single-sensor, single-target tracking, or aided navigation. The recursive Bayes filter is the ideal, nonlinear recursive filter [88]. This general filter is given in the following form and applied recursively:

$$p_{k+1|k}(x_{k+1}|Z^k) = \int p_{k+1|k}(x_{k+1}|x_k)p_k(x_k|Z^k)dx_k, \quad (2.1)$$

$$p_{k+1|k+1}(x_{k+1}|Z^{k+1}) = \frac{p_{k+1}(z_{k+1}|x_{k+1})p_{k+1|k}(x_{k+1}|Z^k)}{p(z_{k+1}|Z^k)}. \quad (2.2)$$

where the denominator in (2.1) is the Bayes normalization constant, $p_{k+1}(z_{k+1}|x_{k+1})$ is the sensor likelihood function, $p_{k+1|k}(x_{k+1}|x_k)$ is the target Markov transition density, $p_k(x_k|Z^k)$ is the posterior distribution conditioned on the data stream $Z^k = z_1, \dots, z_k$, and $p_{k+1|k}(x_{k+1}|Z^k)$ is the time prediction of the posterior to time step $k + 1$. (2.1) is referred to as the Fokker-Planck equation or Chapman-Kolmogorov integral which is equivalent to transition density, for Markovian processes. It describes the time evolution of a probability density function. (2.2) is Bayes' rule.

$$\hat{x}_{k|k}^{MAP} = \arg \max_x p_{k|k}(x|Z^k), \quad (2.3)$$

$$\hat{x}_{k|k}^{EAP} = \int x p_{k|k}(x|Z^k) dx. \quad (2.4)$$

Finally, (2.3) and (2.4) show two ways of obtaining target state estimates from the posterior distribution. The expected a posteriori (EAP) and maximum a posteriori (MAP) estimators minimize the Bayes risk and are commonly used estimators.

Integration operations in (2.2) make the full implementation of the recursive Bayes filter intractable in general conditions. Certain approximations and assumptions are needed in practice. The well-known Kalman filter is a special case of the recursive Bayes filter. Some of these practical filters, which are widely encountered in the literature will be discussed next.

2.1.2. Moment Approximations of the Bayes Filter

Moment approximation is a way to implement the recursive Bayes filter. Assuming the signal-to-noise ratio is large enough [89], the track distributions can be approximated with their first and second-order moments. These moments can be propagated in time using prediction and correction equations of the Bayes filter. Then the Bayes filter can be replaced by the extended Kalman filter (EKF) that propagates these moments after the linearization of either or both process and measurement models. Similarly, when the track distributions are approximated by only the first-order moments, the Bayes filter reduces to the $\alpha - \beta$ filter [89].

2.1.2.1. Kalman Filter

The recursive Bayes filter provides a rigorous, theoretical basis, and optimal tools for single-target state estimation. With a top-down perspective, it is possible to reformulate the Kalman filter as a special case of the recursive Bayes filter. This analysis requires the following fundamental identities about Gaussian densities:

$$\int \mathcal{N}_Q(x - Fx') \cdot \mathcal{N}_P(x' - m) dx' = \mathcal{N}_{Q+FPFT}(x - Fm). \quad (2.5)$$

$$\mathcal{N}_R(r - Hx) \cdot \mathcal{N}_P(x - p) = q(r) \cdot \mathcal{N}_E(x - e). \quad (2.6)$$

where

$$q(r) = \mathcal{N}_{R+HPH^T}(r - Hp), \quad (2.7)$$

$$e = p + K(r - He), \quad (2.8)$$

$$E = (I - KH)P, \quad (2.9)$$

$$K = PH^T(R + HPH^T)^{-1}. \quad (2.10)$$

where P and R are positive-definite square matrices.

Assume that the Markov transition density in the prediction equation (2.1) is a Gaussian density defined as:

$$p_{k+1|k}(x|x') \triangleq \mathcal{N}_{Q_k}(x - F_k x') \quad (2.11)$$

where $p_{k+1|k}(x|x')$ is transition density, F_k is the state transition matrix, Q_k is the covariance matrix, and x' is the previous state vector. Thus, the predicted density in (2.11) is large when x is near $F_k x'$.

Also assume that the prior information ($p_{k|k}(\cdot)$) at time k is defined as a Gaussian density:

$$p_{k|k}(x|Z^k) \triangleq \mathcal{N}_{P_{k|k}}(x - x_{k|k}) \quad (2.12)$$

where $x_{k|k}$ and $P_{k|k}$ is its mean and covariance, respectively. Given (2.11) and (2.12), the Chapman-Kolmogorov integral in (2.1) is

$$\begin{aligned} p_{k+1|k}(x) &\triangleq \int p_{k+1|k}(x|x') \cdot p_{k|k}(x'|Z^k) dx' \\ &= \int \mathcal{N}_{Q_k}(x - F_k x') \cdot \mathcal{N}_{P_{k|k}}(x - x_{k|k}) dx' \end{aligned} \quad (2.13)$$

The argument of the integral contains the product of two Gaussian densities which, in turn, results in another Gaussian density¹:

¹see [86], Appendix D for a formal proof.

$$p_{k+1|k}(x) = \int \mathcal{N}_{Q_k + F_k P_{k|k} F_k^T}(x - F_k x_{k|k}) \cdot \mathcal{N}_E(x' - e) dx' \quad (2.14)$$

$$= \mathcal{N}_{Q_k + F_k P_{k|k} F_k^T}(x - F_k x_{k|k}) \cdot \underbrace{\int \mathcal{N}_E(x' - e) dx'}_{=1} \quad (2.15)$$

$$= \mathcal{N}_{Q_k + F_k P_{k|k} F_k^T}(x - F_k x_{k|k}) \quad (2.16)$$

$$= \mathcal{N}_{P_{k+1|k}}(x - x_{k+1|k}) \quad (2.17)$$

$$= p_{k+1|k}(x|Z^k). \quad (2.18)$$

where $x_{k+1|k} = F_k x_{k|k}$ and $P_{k+1|k} = Q_k + F_k P_{k|k} F_k^T$ are predictor equations of the Kalman filter. Note that, when the prior and transition densities are Gaussian, the Chapman-Kolmogorov integral is equivalent to Kalman state and covariance predictor equations.

In a similar manner, assume that the measurement likelihood function is a Gaussian.

$$p_{k+1}(z|x) \triangleq \mathcal{N}_{R_{k+1}}(z - H_{k+1}x) \quad (2.19)$$

and the predicted density is also a Gaussian

$$p_{k+1|k}(x|Z^k) \triangleq \mathcal{N}_{P_{k+1|k}}(x - x_{k+1|k}) \quad (2.20)$$

Then, the posterior density is constructed using Bayes' rule as in (2.2):

$$\begin{aligned} p_{k+1|k+1}(x) &\triangleq \frac{p_{k+1}(z_{k+1}|x) \cdot p_{k+1|k}(x|Z^k)}{\int p_{k+1}(z_{k+1}|y) \cdot p_{k+1|k}(y|Z^k) dy} \\ &= \frac{\mathcal{N}_{R_{k+1}}(z - H_{k+1}x) \cdot \mathcal{N}_{P_{k+1|k}}(x - x_{k+1|k})}{\int \mathcal{N}_{R_{k+1}}(z - H_{k+1}y) \cdot \mathcal{N}_{P_{k+1|k}}(y - x_{k+1|k}) dy} \end{aligned} \quad (2.21)$$

where H_{k+1} is the measurement matrix and R_{k+1} is measurement covariance. The numerator of (2.21) describes how the new measurement modifies the prior information and the denominator is the normalization term. If we use the Gaussian identity theorem as in the predictor step, we get the following expressions:

$$p_{k+1|k+1}(x) = \frac{\mathcal{N}_{D_{k+1}}(z_{k+1} - H_{k+1}x_{k+1|k})\mathcal{N}_E(x - e)}{\mathcal{N}_{D_{k+1}}(z_{k+1} - H_{k+1}x_{k+1|k}) \int \mathcal{N}_E(y - e)dy} \quad (2.22)$$

$$= \mathcal{N}_E(x - e) \quad (2.23)$$

$$= \mathcal{N}_{P_{k+1|k+1}}(x - x_{k+1|k+1}) \quad (2.24)$$

$$= p_{k+1|k+1}(x|Z^{k+1}). \quad (2.25)$$

where the following expressions are the result of applying the Gaussian identity theorem

$$\begin{aligned} P_{k+1|k+1}^{-1} &= E^{-1} \triangleq P_{k+1|k}^{-1} + H_{k+1}^T R_{k+1}^{-1} H_{k+1} \\ P_{k+1|k+1}^{-1} x_{k+1|k+1} &= E^{-1} e \triangleq P_{k+1|k}^{-1} x_{k+1|k} + H_{k+1}^T R_{k+1}^{-1} z_{k+1} \\ D_{k+1} &\triangleq R_{k+1} + H_{k+1} P_{k+1|k} H_{k+1}^T \end{aligned}$$

Thus, the posterior density $f_{k+1|k+1}(x|Z^{k+1})$ at time $k+1$ is also found to be Gaussian with mean $x_{k+1|k+1}$ and covariance $P_{k+1|k+1}$. In the above equations, Kalman filter correction for equations can be easily identified. Consequently, when the prior density and the measurement likelihood are Gaussian densities, the recursive Bayes filter correction is equivalent to the Kalman correction step.

After applying the Bayes' rule and obtaining the posterior density as a Gaussian distribution, it is required to extract the state and covariance estimates from the posterior distribution. This can be done by calculating the expectations (EAP) as follows:

$$\int x \cdot \mathcal{N}_{P_{k+1|k+1}}(x - x_{k+1|k+1}) dx = x_{k+1|k+1} \quad (2.26)$$

$$\int x x^T \cdot \mathcal{N}_{P_{k+1|k+1}}(x - x_{k+1|k+1}) dx = P_{k+1|k+1}. \quad (2.27)$$

Thus, we have derived the Kalman filter prediction and correction equations from the recursive Bayes filter formulation, with the following assumptions:

- Initial prior information, Markov transition density, and the measurement likelihood are Gaussian distributions.
- The time evolution of the state vector is linear, represented by the transition matrix, F .
- The measurement function relating the observation vector to the state vector is a linear mapping, represented by the measurement matrix, H .

Under these assumptions, the recursive Bayes filter has a closed-form solution and reduces to the well-known Kalman filter.

2.1.2.2. Extended Kalman Filter

The Kalman filter, regardless of how widely it is used, is too restrictive in real-world applications. When either or both of the measurement and process (or motion) models are not linear, the Kalman filter is not applicable and, at best, some approximations are needed.

The so-called EKF linearizes the process and/or measurement models, $f(\cdot)$ and $h(\cdot)$, respectively, using a Taylor series expansion. Essentially, the main assumption here is that these functions be *locally linear*, instead of being globally linear. The linearization can be done by truncating the Taylor series expansion and discarding the higher-order terms. Thus, the predicted and posterior densities can be approximated by a single Gaussian.

Assume that at any time k , the non-linear Gaussian process, and measurement models are given as

$$x_{k+1|k} = f_{k+1|k}(x_k, v_k) \quad (2.28)$$

$$z_{k+1} = h_{k+1}(x_{k+1}, w_{k+1}) \quad (2.29)$$

where v_k and w_{k+1} are zero-mean Gaussian noise variables. $x_k \in \mathbb{R}^{n_x}$ is the random state vector, $z_k \in \mathbb{R}^{n_z}$ is the measurement vector, $f : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$, and $h : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_z}$. If we expand f_k to first order around $x_{k|k}$, the state transition matrix will be defined by

$$F_k \triangleq \frac{\partial f_k}{\partial x}(x, 0) \Big|_{x=x_{k|k}}, \quad G_k \triangleq \frac{\partial f_k}{\partial v}(x_{k|k}, v) \Big|_{v=0} \quad (2.30)$$

where the partial derivative is a gradient derivative and G_k is the input matrix. Similarly, for the measurement function,

$$H_k \triangleq \frac{\partial h_k}{\partial x}(x, 0) \Big|_{x=x_{k+1|k}}, \quad U_k \triangleq \frac{\partial h_k}{\partial w}(x_{k|k}, w) \Big|_{w=0} \quad (2.31)$$

After the linearization step, the same prediction and correction equations are applied as the standard linear Kalman filter.

Because only the first-order terms are used and the higher-order terms are discarded, when the non-linearities in the process and/or measurement models are severe, the EKF performs poorly. Also, if the functions are not continuous the EKF is not applicable [21].

2.1.2.3. Unscented Kalman Filter

Because of the divergence problem of the EKF approach when the process and/or measurement functions have higher-order non-linearities, the applicability of the EKF is limited. An alternative approach to this is the UKF where, instead of linearizing the process and measurement functions, a deterministic sampling principle, the so-called unscented transform (UT) is employed to approximate predicted and posterior densities. The UT is used to propagate first and second-order moments of the predicted and posterior densities through these nonlinear functions. In a sense, this process represents a *statistical linearization*. These densities are approximated using a deterministically selected point set, referred to as sigma points, $\mathcal{X}^{(i)}$, and their respective weights $\omega^{(i)}$.

Suppose that, at time k , the density (either predicted or posterior) is represented by a set of sigma points and associated weights $\{\mathcal{X}_k^{(i)}, \omega^{(i)}\}_{i=1:N}$, where $N = 2n_x + 1$, chosen according to the following

$$\mathcal{X}_k^{(i)} = \begin{cases} x_k, & i = 1 \\ x_k + [\sqrt{(n_x + \kappa P_k)}]_{i-1}, & i = 2, \dots, n_x + 1 \\ x_k - [\sqrt{(n_x + \kappa P_k)}]_{i-n_x-1}, & i = n_x + 2, \dots, 2n_x + 1 \end{cases} \quad (2.32)$$

$$\omega^{(i)} = \begin{cases} \frac{\kappa}{n_x + \kappa}, & i = 1 \\ \frac{1}{2(n_x + \kappa)}, & i = 2, \dots, 2n_x + 1 \end{cases} \quad (2.33)$$

where κ is the scaling constant. $[Q]_i$ denotes the i -th row of the matrix Q .

Then, the predicted density can be propagated through its mean and covariance as below:

$$x_{k+1|k} = \sum_{i=1}^N \omega^{(i)} p_{k+1|k}(\mathcal{X}_k^{(i)}), \quad (2.34)$$

$$P_{k+1|k} = \sum_{i=1}^N \omega^{(i)} [p_{k+1|k}(\mathcal{X}_k^{(i)}) - x_{k+1|k}] [p_{k+1|k}(\mathcal{X}_k^{(i)}) - x_{k+1|k}]^T + Q_k. \quad (2.35)$$

Similarly, the posterior density after the non-linear measurement update step is

$$z_{k+1|k} = \sum_{i=1}^N \omega^{(i)} h_{k+1}(\mathcal{X}_{k+1|k}^{(i)}), \quad (2.36)$$

$$x_{k+1|k+1} = x_{k+1|k} + K_{k+1}(z_{k+1} - z_{k+1|k}), \quad (2.37)$$

$$P_{k+1|k+1} = P_{k+1|k} - K_{k+1} S_{k+1} K_{k+1}^T, \quad (2.38)$$

$$K_{k+1} = P_{xz} S_{k+1}^{-1}, \quad (2.39)$$

$$S_{k+1} = P_{zz} + R_{k+1}, \quad (2.40)$$

$$P_{xz} = \sum_{i=1}^N \omega^{(i)} [\mathcal{X}_{k+1|k}^{(i)} - x_{k+1|k}] [h_{k+1}(\mathcal{X}_{k+1|k}^{(i)}) - z_{k+1|k}]^T, \quad (2.41)$$

$$P_{zz} = \sum_{i=1}^N \omega^{(i)} [h_{k+1}(\mathcal{X}_{k+1|k}^{(i)}) - z_{k+1|k}] [h_{k+1}(\mathcal{X}_{k+1|k}^{(i)}) - z_{k+1|k}]^T \quad (2.42)$$

where $z_{k+1|k}$ is the predicted measurement.

In the UKF approach, unlike the EKF, the Jacobians are not required. This makes it possible to apply the UKF to the problems with discontinuous functions [98]. The accuracy of the transformed function is up to the second-order Taylor series or third-order if the random variable is Gaussian.

2.1.3. Density Approximation of the Bayes Filter

Up until this point, the moment approximation methods to the recursive Bayes filter are summarized. The main idea is that instead of full predicted/posterior densities, their respective first and second-order moments can be propagated. However, they either assume that the models are linear or random variables are Gaussian distributed to be calculated in closed form. Regardless of how widely used, they can not provide sufficient flexibility in general, because real-world problems are seldom linear and Gaussian.

An alternative approach to these methods is to approximate the density itself by numerical methods. This provides a much more flexible modeling capability because there is no assumption of linearity or Gaussianity. This can be done through sequential Monte Carlo (SMC) approximation, also known as particle filtering framework.

2.1.3.1. Particle Filter

SMC approximation, conceptually, can be considered a generalization of the UKF method. In SMC, the posterior distribution is represented as a set of vectors, called particles and their associated weights [86] which represent the importance of the particle in the system. This particle system is represented as $\{x^{(i)}, w^{(i)}\}_{i=1}^N$, with $\sum_i w^i = 1$. Suppose that the posterior density $p_{k|k}(x|Z^k)$ at time k is given. Mathematically, this can be represented as the following approximation:

$$\int \theta(x) \cdot p_{k|k}(x|Z^k) dx \cong \sum_{i=1}^N w_{k|k}^i \cdot \theta(x_{k|k}^i) \quad (2.43)$$

where $\theta(x)$ is a unitless function. The particles are random samples drawn from the posterior distribution:

$$x_{k|k}^1, \dots, x_{k|k}^N \sim p_{k|k}(\cdot|Z^k). \quad (2.44)$$

The weights, as well as particles, are propagated through recursive Bayes filtering equations. Assume that the posterior is approximated as

$$p_{k|k}(x|Z^k) \cong \frac{1}{N} \sum_{i=1}^N \delta_{x_{k|k}^i}(x). \quad (2.45)$$

where $\delta_{x_i}(x)$ is the Dirac delta density concentrated at x_i . Suppose that, for each $x_{k|k}^i$, ν random samples have been drawn from the Markov transition density:

$$x_{k+1|k}^{1,i}, \dots, x_{k+1|k}^{\nu,i} \sim p_{k+1|k}(\cdot|x_{k|k}^i, Z^k). \quad (2.46)$$

Then, the Markov transition density can be approximated as

$$p_{k+1|k}(x|x_{k|k}^i) \cong \frac{1}{\nu} \sum_{j=1}^{\nu} \delta_{x_{k+1|k}^{j,i}}(x). \quad (2.47)$$

If we substitute (2.45) and (2.47) into (2.1), we get

$$\int \theta(x) \cdot p_{k+1|k}(x|Z^k) dx \cong \int \int \theta(x) \cdot p_{k+1|k}(x|x') \cdot p_{k|k}(x'|Z^k) dx' dx \quad (2.48)$$

$$= \frac{1}{N} \sum_{i=1}^N \int \theta(x) \cdot p_{k+1|k}(x|x_{k|k}^i) dx \quad (2.49)$$

$$= \frac{1}{N\nu} \sum_{j=1}^{\nu} \sum_{i=1}^N \theta(x_{k+1|k}^{j,i}). \quad (2.50)$$

ν is the number of samples that are drawn from the predicted density, for each particle and is usually set $\nu = 1$. Then,

$$\int \theta(x) \cdot p_{k|k}(x|Z^k) dx \cong \frac{1}{N} \sum_{i=1}^N \theta(x_{k+1|k}^i). \quad (2.51)$$

In this approach, the importance density has been simply chosen to be dynamic prior [86]. However, designing the importance density (or proposal density) is an important step in SMC implementations.

Substituting (2.45) into the denominator of (2.2), yields

$$p_{k+1}(z_{k+1}|Z^k) = \int p_{k+1}(z_{k+1}|x) \cdot p_{k+1|k}(x|Z^k) dx \quad (2.52)$$

$$\cong \frac{1}{N} \sum_{i=1}^N p_{k+1}(z_{k+1}|x_{k+1|k}^i). \quad (2.53)$$

Then the Bayes measurement update equation becomes

$$\int \theta(x) \cdot p_{k+1|k+1}(x|Z^{k+1}) dx = \frac{\int \theta(x) \cdot p_{k+1}(z_{k+1}|x) \cdot p_{k+1|k}(x|Z^k) dx}{p_{k+1}(z_{k+1}|Z^k)} \quad (2.54)$$

$$\cong \frac{\frac{1}{N} \sum_{i=1}^N \theta(x_{k+1|k}^i) \cdot p_{k+1}(z_{k+1}|x_{k+1|k}^i)}{\frac{1}{N} \sum_{e=1}^N p_{k+1}(z_{k+1}|x_{k+1|k}^e)} \quad (2.55)$$

$$= \sum_{i=1}^N w_{k+1|k+1}^i \cdot \theta(x_{k+1|k+1}^i). \quad (2.56)$$

where $x_{k+1|k+1}^i \triangleq x_{k+1|k}^i$ and

$$w_{k+1|k+1}^i \triangleq \frac{p_{k+1}(z_{k+1}|x_{k+1|k}^i)}{\sum_{e=1}^N p_{k+1}(z_{k+1}|x_{k+1|k}^e)}. \quad (2.57)$$

Note that, after the measurement update step in (2.56) and (2.57), the newly updated particles have now unequal weights. Some of the particles have very small weights while some have large weights. To eliminate the ones with smaller weights and duplicate the ones with larger weights, a resampling scheme is employed. This is because, after a few update steps, most of the particles would have diminishing weights and would not be contributing to the approximation and only a few particles would be used in the approximation. Then most of the particles would be useless.

Since the resampling step creates identical copies of the particles, it causes a phenomenon known as *particle degeneracy*. The statistical diversity of the resampled particles should be increased to mitigate this effect².

After the posterior has been constructed using the particle system, the state estimation can be done via EAP estimation:

$$\begin{aligned}\hat{x}_{k+1|k+1}^{EAP} &= \int x \cdot p_{k+1|k+1}(x|Z^{k+1})dx \\ &\cong \frac{1}{N} \sum x_{k+1|k+1}^i.\end{aligned}\tag{2.58}$$

Note that, because of the multi-modal structure of the approximated posterior density, for SMC filters the MAP estimation is more difficult. To use MAP estimation, first, the significant modes of the posterior distribution must be estimated. Then, assuming the posterior is approximated as a weighted sum of Gaussians, the MAP estimation can be applied to find the parameters.

2.2. Multi-Target Tracking

The previous section has summarized single-target tracking from the Bayesian point of view and reviewed the most widely used methods. The recursive Bayes filter has been presented and has been shown that it provides the theoretical foundation for Bayesian data fusion. The general construct of the conventional MTT is shown in Fig. 2.1.

Despite the multi-target tracking (MTT) problem not being a direct generalization of single-target tracking in general, the (single-target) Bayes filter recursion has a multi-target correspondence: The multi-target Bayes filter. In the multi-target Bayes filter, single-target densities turn into multi-target densities and

²More inclined readers can refer to [86] and the references therein.

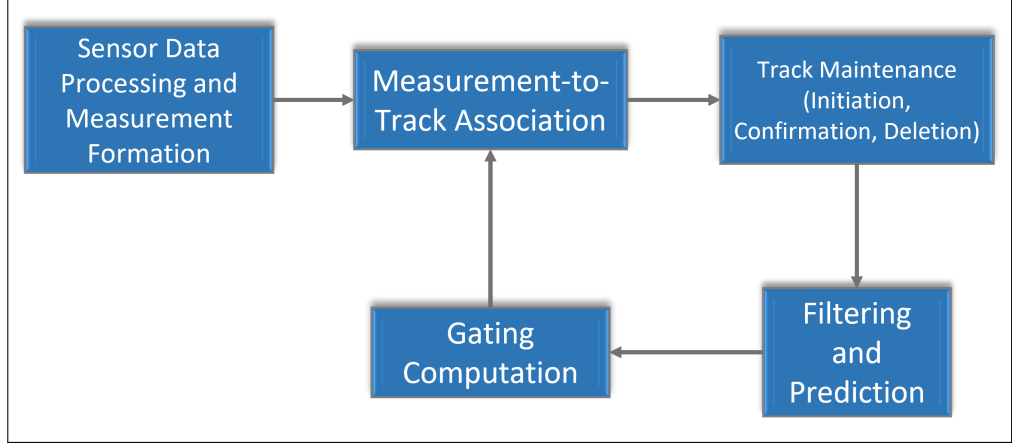


Figure 2.1: Conventional MTT architecture.

the conventional integral operation turns into multi-target integration. For the sake of completeness, the multi-target Bayes filter is introduced here first.

Let $Z^{(k)} : Z_1, \dots, Z_k$ be the time sequence of measurement sets, and $p_{k+1|k}(X|X')$ and $p_{k+1}(Z|X)$ be the *multi-target Markov density* and multi-target likelihood function, respectively. Then the multi-target Bayes filter is described by the following expressions.

$$p_{k+1|k}(X|Z^{(k)}) = \int p_{k+1|k}(X|X') \cdot p_{k|k}(X'|Z^{(k)}) \delta X' \quad (2.59)$$

$$p_{k+1|k+1}(X|Z^{(k+1)}) = \frac{p_{k+1}(Z_{k+1}|X)p_{k+1|k}(X|Z^{(k)})}{p(Z_{k+1}|Z^{(k)})} \quad (2.60)$$

where the integral in (2.59) is a set integral (see Section 2.3.1.).

When propagated through time, the multi-target Bayes filter provides a unified, systematic, and probabilistic approach to MTT [86]. However, in general, as in the case of its single-target counterpart, it is intractable to calculate the exact multi-target posterior in multi-target Bayes filter form, and approximations are necessary to formulate practical algorithms [98].

As mentioned above, MTT is not the direct generalization of single-target tracking. While the single-target tracking methods are the building blocks of the MTT, it is an entirely distinct discipline with important sub-problems. In this section, we briefly summarize the critical elements of the (conventional) MTT and mostly focus on methods that attempt to solve them. The main problem in MTT is the so-called *data association uncertainty* which arises from the ambiguity between target tracks and measurements at a given time instance. When there are multiple measurements, some of which are possibly target-originated, and the remaining are false alarms (or clutter),

there is an ambiguity such that the measurement which will update a specific target track is not known beforehand. Moreover, there is no knowledge that whether or not a specific target track actually generated any measurement. Thus, the first thing needed to do is to statistically associate the existing measurements with existing target tracks and eliminate the possible false alarms. This process is called measurement-to-track association (M2TA) and is a crucial step in MTT applications.

2.2.1. Measurement-to-Track Association (M2TA)

The focus of this section stems from the following question [86]: *At an arbitrary time, how multiple observations of a sensor and multiple existing tracks are related? Which of the multiple measurements acquired originated from a specific target track, if any?* Suppose that, at time t_k , there are n triplets: $(l^1, x_{k|k}^1, P_{k|k}^1), \dots, (l^n, x_{k|k}^n, P_{k|k}^n)$. For the i -th track, x^i is the state vector, P^i is its covariance matrix, and l^i is the track label. The Gaussian distribution $p_i(x) = \mathcal{N}_{P_{k|k}^i}(x - x_{k|k}^i)$ is called track density of i -th track.

Next, suppose that at time t_{k+1} , m measurements $Z = \{z_1, \dots, z_m\}$ are acquired. Since there might be clutter, $m > n$ in general. In order to obtain the next track estimates, we should determine which tracks will be updated with which measurement. For this purpose, an association hypothesis is constructed, that is, $\mathcal{H}$: for each track i , the target $(l^i, x_{k|k}^i, P_{k|k}^i)$ generated the detection $z_{\tau(i)}$. The remaining $m - n$ measurements are considered false alarms or previously undetected targets. The hypothesis $\mathcal{H}$ is an M2TA. Among all possible associations, an optimum association must be found. This is typically done by calculating the distance of each observation to all targets. Until recently, most of the multi-target tracking algorithms were M2TA-based, i.e. MHT and JPDA Filter.

Mahler discussed the interpretation of M2TA's in [32] and deduced that M2TA's are heuristic approximations of the multi-target Bayes filter, moreover, he argued the physical meaning of an M2TA. He claimed the M2TA approach is not consistent with Bayesian statistics, which Bayes' rule and therefore recursive Bayes filter is based on.

By the beginning of the 2000s, Mahler pioneered the RFS theory by introducing the PHD filter, an RFS-based first-order approximation filter [99]. RFS-based filters are able to handle M2TA or data association problems mentioned in previous paragraphs. Basically, the RFS-based filters present a workaround to the data

association problem. Moreover, some RFS-based filters do not require M2TA and focus on finding optimal (or sub-optimal) estimates of the multi-target state [85].

In the remainder of this section, we will briefly summarize the widely used MTT approaches that attempt to solve mainly the M2TA problem.

2.2.2. Joint Probabilistic Data Association (JPDA)

The JPDA filter is an approach that aims to solve the data association uncertainty or M2TA in MTT. It is an extended form of the PDA filter for single target tracking applications. However, its high computational complexity is its main drawback, and most approximations trying to mitigate that have been heuristic techniques and sacrifice accuracy for computational tractability [25]. In this section, the JPDA approach is briefly reviewed as it is one of the classical approaches for MTT.

Let $x_k^1, \dots, x_k^N$ and $z_k^1, \dots, z_k^M$ be sets of N target states and M observed measurements at time k , respectively. Let $p(d_i^j = 1)$ denote the data association probability between a measurement index j and track index i . It is assumed that the measurement j originated from the target i . Assuming a linear Gaussian model, $p(d_i^j)$ is obtained as follows:

$$p(d_i^j) \propto \begin{cases} (1 - p_D)\beta & i = 0, \\ p_D \mathcal{N}(z^j; \hat{x}^j, \Sigma_S) & \text{otherwise,} \end{cases} \quad (2.61)$$

where $\hat{x}^j$ is the j -th predicted target position, p_D is the detection probability, and β is the clutter density.

Suppose that Θ is formed by all the possible combinations of M2TA satisfying (i) each measurement is assigned to a single track (at most), and (ii) each track can only be assigned a single measurement³. Mathematically, this can be represented as

$$\begin{aligned} \Theta = \{ \theta = (d_i^j)_{i \in [M], j \in [N]} \mid d_i^j \in \{0, 1\} \\ \wedge \sum_{j=1}^N d_i^j \leq 1, \forall i \in [M] \wedge \sum_{i=0}^M d_i^j = 1, \forall j \in [N] \}. \end{aligned} \quad (2.62)$$

where $\theta \in \Theta$ is a binary vector and Θ is the space of all possible combinations of binary assignments. The JPDA probability $q(d_i^j)$ is marginalized as

³Note that these conditions are the main assumptions of the standard measurement model which is described in both 2.1. and 3.2.

$$q(d_i^j) = \sum_{\theta \in \Theta_i^j} p(\theta) \quad (2.63)$$

where $\Theta_i^j = \{\theta \in \Theta | d_i^j = 1\}$ and

$$p(\theta) = \prod_{\substack{\forall r \in [M]_0 \\ \forall k \in [N]}} (p(d_r^k))^{d_r^k}. \quad (2.64)$$

Then, the joint association probabilities $q(d_i^j)$ are normalized and used to update the j -th target state. The JPDA performs the standard Kalman prediction step. The Kalman update uses the information from all the measurements, weighted with association probabilities. With the use of joint association probabilities, the JPDA tries to avoid conflicting M2TA hypotheses.

Although it provides an intuitive alternative, there are several shortcomings. First, it is assumed that the number of targets is fixed and known beforehand. This is not always a reasonable assumption since the scene can be very dynamic. Second, the computational complexity grows exponentially with both the number of targets and the number of measurements, so some pruning step is necessary to eliminate the unlikely hypotheses.

2.2.3. Multiple Hypothesis Tracking (MHT)

Another classical approach to MTT is the celebrated *multiple hypothesis tracking* (MHT) approach of Reid [28]. In MHT, the conflicting association hypotheses are mitigated by forming alternative hypotheses at previous time steps for all possible combinations. Then the set of hypotheses is propagated through time expecting the association uncertainties to be resolved with future measurements.

There are two variations to the MHT approach: (i) hypothesis-oriented MHT (HO-MHT) and (ii) track-oriented MHT (TO-MHT). The implementation architectures of the two variations are shown in Fig. 2.2a and 2.2b. In this section, we will present a summary of the basics of the MHT approach.

Suppose that at a given time step, there are two target tracks present T_1 and T_2 and respective predicted target positions P_1 and P_2 , as shown in Fig. 2.3. Together, they represent a hypothesis: $\mathcal{H}_1$. Now, assume that three measurements have been obtained during the observation: O_1, O_2, O_3 . Given the initial hypothesis and the measurements,

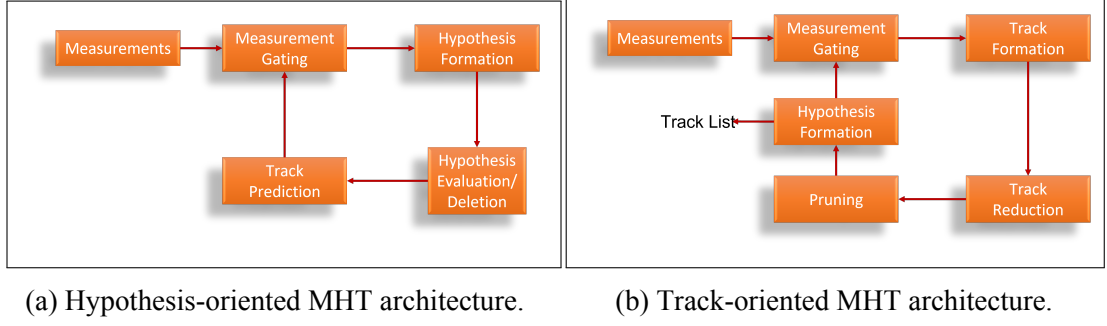


Figure 2.2: Two MHT variations in implementation.

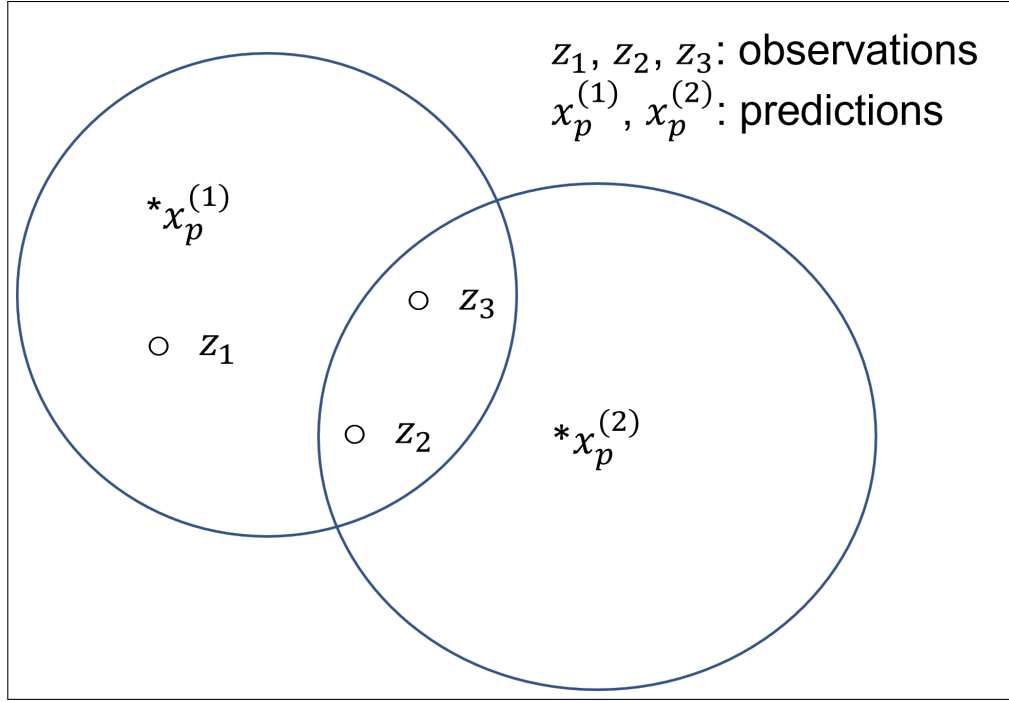


Figure 2.3: An example of data association uncertainty with two tracks and three observations.

there would be 10 feasible hypotheses that can be generated. One likely hypothesis, according to Fig. 2.3, is that T_1 is updated with O_1 , and T_2 is updated with either O_2 or O_3 . Another hypothesis, however unlikely, would be that none of the observations originated from existing tracks, thus implying new targets or false alarms. Therefore, T_1 and T_2 are not updated, and the observations initiate new tracks. Upon receiving new measurements, each hypothesis generates a new set of hypotheses considering all possible M2TA.

A likelihood ratio for the track and hypothesis evaluation is calculated, which represents the likelihood of initiating a track from the given data (prior and evidence), as follows:

$$LR = \frac{p(D|\mathcal{H}_1)P_0(\mathcal{H}_1)}{p(D|\mathcal{H}_0)P_0(\mathcal{H}_0)} \triangleq \frac{P_T}{P_F} \quad (2.65)$$

where $\mathcal{H}_1$ and $\mathcal{H}_0$ are the target and false alarm hypotheses with probabilities P_T and P_F , respectively, and D is the data. The LR score is used to initiate and terminate the tracks when it reaches a certain threshold.

2.3. Random Finite Sets (RFS)

A random variable usually has a fixed dimension, such as a vector whose elements are random. A random finite set (RFS) is a set whose cardinality (i.e. number of elements) as well as elements are random valued variables. The RFS concept is especially very well suited for MTT problems in which the target states and the number of targets both randomly vary. Another distinct feature of the RFS concept is that the elements are distinct and unordered [98], while in a vector form, the elements are ordered scalar variables. A random set, therefore, has two quantities: a discrete cardinality distribution on non-negative integers and for each cardinality, a joint probability distribution. A more general form of this concept is called a *point process* which was suggested for MTT by Mahler [10], [86]. Since the RFSs are not conventional vector-valued variables, the standard calculus formulas are not directly applicable. The so-called finite set statistics (FISST) framework provides the necessary tools for dealing with the RFS variables [99], [100], and at the center of this framework is the so-called set integral which will be described briefly, for the sake of completeness.

2.3.1. FISST Multi-Target Calculus

In this section, we will provide a brief overview of the FISST calculus for the multi-target tracking (MTT) problem. Specifically, we will describe the belief functions that represent the multi-target motion and sensor models [86], [101], [102]. These belief functions will serve as the primary means for characterizing the multi-target motion and sensor models. We will then use set derivatives and set integrals to calculate the multi-target Markov density and likelihood functions. The multi-target sensor likelihoods and Markov densities, similar to their single-target counterparts, will be utilized in the multi-target Bayes filter. The notation used in

this section will follow the conventions established by Mahler in [86].

A multi-target system state is described as a subset $X = \{x_1, \dots, x_n\} \subseteq \mathfrak{X}$ where $\mathfrak{X}$ is a subset of Euclidean space with $X = \emptyset$ for $n = 0$. Similarly, the sensor measurement set is a subset of measurement space defined as $Z = \{z_1, \dots, z_m\} \in \mathfrak{Z}$ with $Z = \emptyset$ for $m = 0$. Then the unknown, random multi-target state and measurement sets are represented as *random finite set (RFS)* $\Xi \subseteq \mathfrak{X}$ and $\Sigma \subseteq \mathfrak{Z}$, respectively. Thus, a random finite set is a set variable where its elements as well as the number of elements it contains, i.e. its cardinality, are random variables.

2.3.1.1. Set integral

The statistics of a set variable X are described by a multi-target density function $f(X) \geq 0$. The set integral, similar to the standard probability density function, of $f(X)$ is

$$\int f(X) \delta X = f(\emptyset) + \sum_{n \geq 1} \frac{1}{n!} \int f_n(x_1, \dots, x_n) dx_1 \dots dx_n \quad (2.66)$$

where $f_n(x_1, \dots, x_n) = f(\{x_1, \dots, x_n\})/n!$ and units of measurement of $f(X)$ are $K^{|X|}$ and K is the unit of measurement of $\mathfrak{X}$. Similar to its single target counterpart $\int f_\Xi(X) \delta X = 1$. The cardinality distribution of Ξ is

$$p_\Xi(n) = \Pr(|\Xi| = n) = \int_{|X|=n} f_\Xi(X) \delta X \quad (2.67)$$

$$= \frac{1}{n!} \int f_\Xi(\{x_1, \dots, x_n\}) dx_1 \dots dx_n. \quad (2.68)$$

2.3.1.2. Probability generating functionals

The probability-generating functionals (p.g.fl.) are used to transform multi-target densities into simpler/useful expressions. Especially in the Bayes filtering equations, e.g. prediction integral (2.59), the problem can be transformed into an equation in the following form

$$G_{k+1|k}[h] = G_{k|k}[T[h]] \quad (2.69)$$

where $G_{k|k}[h]$ is the p.g.fl. of the previous multi-target posterior distribution $f_{k|k}(X|Z^{(k)})$ and $G_{k+1|k}[h]$ is the p.g.fl. of predicted multi-target distribution $f_{k+1|k}(X|Z^{(k)})$. Thus, $T[h]$ represents a functional transformation of the multi-target motion model.

The p.g.fl. of Ξ for test functions $0 \leq h(x) \leq 1$ is defined as

$$G_{\Xi}[h] = \int h^X f_{\Xi}(X) \delta X \quad (2.70)$$

where

$$h^X = \begin{cases} 1, & X = \emptyset \\ \prod_{x \in X} h(x), & o.w. \end{cases} \quad (2.71)$$

The power functional h^X satisfies the generalized binomial theorem:

$$(h_1 + h_2)^X = \sum_{Y \subseteq X} h_1^Y h_2^{X-Y}. \quad (2.72)$$

2.3.1.3. Functional derivative

The intuitive definition of the functional derivative of $G_{\Xi}[h]$ is [86]

$$\frac{\delta G_{\Xi}}{\delta x}[h] = \lim_{\varepsilon \rightarrow 0} \frac{G_{\Xi}[h + \varepsilon \delta_x] - G_{\Xi}[h]}{\varepsilon} \quad (2.73)$$

where $\delta_x(y)$ is the Dirac delta functional concentrated at x . For $X = \{x_1, \dots, x_n\}$ with $|X| = n$ then

$$\frac{\delta G_{\Xi}}{\delta X}[h] = \frac{\delta^n G_{\Xi}}{\delta x_1 \cdots \delta x_n}[h] \quad (2.74)$$

$$= \frac{\delta}{\delta x_n} \frac{\delta^{n-1} G_{\Xi}}{\delta x_1 \cdots \delta x_{n-1}}[h] \quad (2.75)$$

For example, the p.g.fl. and multi-target distribution of an RFS are related by

$$f_{\Xi}(X) = \frac{\delta G_{\Xi}}{\delta X}[0]. \quad (2.76)$$

and the probability hypothesis density (PHD) of Ξ is

$$D_\Xi(x) = \int f_\Xi(\{x\} \cup X) \delta X = \frac{\delta G_\Xi}{\delta x}[1]. \quad (2.77)$$

The multi-target Bayes filter (2.59) and (2.60) can be represented in terms of p.g.fl.s for standard multi-target motion and measurement models, respectively, as follows:

$$G_{k+1|k}[h|X'] = \int h^X f_{k+1|k}(X|X') \delta X = G_B^{k+1|k}[h] \cdot (1 - p_S^{k+1|k} + p_S^{k+1|k} M_h^{k+1|k})^{X'} \quad (2.78)$$

$$G_{k+1}[g|X] = \int g^Z f_k(Z|X) \delta Z = e^{\kappa_k[g-1]} \cdot (1 - p_D^k + p_D^k L_g^k)^X. \quad (2.79)$$

where $p_D(x)$ is the probability of detection of the sensor, $L_g^k(x) = \int g(z) f_k(z|x) dz$, $\kappa_k(z)$ is the intensity of Poisson clutter process, $p_S(x')$ is the probability of survival of a target, $f_{k+1|k}(x|x') = M_x^{k+1|k}(x')$ is the Markov density, $G_B^{k+1|k}[h]$ is the p.g.fl. of multi-target birth RFS, and $M_h^{k+1|k}(x') = \int h(x) f_{k+1|k}(x|x') dx$.

Then the p.g.fl. form of (2.59) and (2.60) are as follows:

$$G_{k+1|k}[h|Z^{(k)}] = G_B^{k+1|k}[h] \cdot G_{k|k}[1 - p_S + p_S M_h|Z^{(k)}], \quad (2.80)$$

$$G_{k+1|k+1}[h|Z^{(k)}] = \frac{\frac{\delta F_k}{\delta Z_k}[0, h]}{\frac{\delta F_k}{\delta Z_k}[0, 1]}. \quad (2.81)$$

where $F_k[g, h] = e^{\kappa_k[g-1]} \cdot G_{k+1|k}[h(1 - p_D + p_D L_g)|Z^{(k)}]$.

2.3.2. Random Finite Set-based Filters

The Random Finite Set (RFS) theory, a concept rooted in the point process theory formulated by Moyal over half a century ago, presents itself as a particular instance of the more comprehensive notion of point processes [33]. By combining RFS with finite set statistics (FISST), a robust theoretical framework is established to address the challenges associated with multi-sensor, multi-target tracking, and expert system issues [103].

In this context, the representation of targets, measurements, and sensor suites - which are comprised of individual sensors - as random sets, allows for the number

of these entities to be treated as random variables alongside their respective states. This application of random finite set modeling paves the way for the construction of multi-sensor, multi-target problems in a mathematically rigorous manner, thereby facilitating the derivation of principled statistical approximations for the more general multi-target nonlinear Bayes filter.

Utilizing the Finite Set Statistics (FISST) multi-object calculus, which encompasses the multi-object equivalents of the renowned Chapman-Kolmogorov prediction integral and Bayes update, it becomes feasible to establish *true* multi-target likelihood functions and *true* multi-target Markov densities. As a result of the veracity inherent in these multi-target motion and measurement models, exact approximations of multi-sensor, multi-target filters can be devised through the application of FISST multi-object calculus [33].

This FISST calculus is equipped with essential tools, such as set integration, set differentiation, and functional derivatives, which enable the computation of both predicted and posterior multi-object densities. Consequently, the FISST multi-object calculus offers a comprehensive framework for addressing complex problems in multi-sensor, multi-target tracking.

Consequently, the Random Finite Set (RFS) framework provides a comprehensive toolkit that facilitates the application of multi-sensor, multi-target integral, and differential calculus to intricate statistical estimation challenges. This framework enables the derivation of formal multi-sensor, multi-target motion, and measurement models in a structured manner. Furthermore, it empowers practitioners to harness expert system concepts, such as fuzzy logic, Dempster-Shafer theory, and other inference rules, for the development of principled algorithms within a mathematically rigorous framework. This integration of expert systems with the RFS framework promotes the advancement of robust, sophisticated solutions for multi-sensor, multi-target tracking, and estimation problems.

In the following sections, several well-known RFS-based multi-target filters, namely PHD, GLMB, and PMBM are presented. Important multi-object distributions and random finite sets are presented in the appendix 2.7.2.-2.7.2..

2.3.2.1. Probability Hypothesis Density (PHD) Filter

One of the fundamental and earliest methods using the RFS framework is the celebrated PHD filter [99], [104]. The prior assumption is that the SNR level is high enough to be considered a sufficient statistic. Meaning that the higher-order moments of the multi-target posterior distribution become less significant, thus only the first-order moment of the posterior, i.e. the PHD, can be considered a sufficient statistic. Therefore, an analogy to the single-target $\alpha - \beta$ filter can be established as it is the first-order approximation as well.

The PHD filter is a tractable approximation to the multi-target Bayes filter. It is derived with the assumption that the predicted multi-target distribution is a Poisson process. To approximate the multi-target Bayes filter, a definition of the *expected value* over the RFSs is needed since the standard definition does not apply to the set-theoretic concept. Mahler presents this definition [86] as follows:

$$\mathbb{E}[T_\Xi] \triangleq \int T_X \cdot f_\Xi(X) \delta X. \quad (2.82)$$

where the integral is a set integral and T_X is a transformation that maps a finite-set X to a vector T_X . A common choice is $T_X = \delta_X$ where

$$\delta_X(x) = \begin{cases} 0, & X = \emptyset \\ \sum_{w \in X} \delta_w(x), & o.w. \end{cases} \quad (2.83)$$

Thus,

$$D_\Xi(x) \triangleq \mathbb{E}[\delta_\Xi(x)] = \int \delta_X(x) f_\Xi(X) \delta X \quad (2.84)$$

is the multi-target expected value of a multi-target distribution and called the probability hypothesis density (PHD) or the *intensity function* of Ξ [86].

The integral of $D_\Xi(x)$ in a region S gives the expected number of targets in the region

$$\int_S D_\Xi(x) dx = \mathbb{E}[|S \cap \Xi|]. \quad (2.85)$$

Consequently, $D_\Xi(x)$ can be interpreted as the *target density* at x , in other words, it gives the expected number of targets in a hyper-volume [86].

In PHD filter, it is assumed that the multi-target posterior distribution is a multi-target Poisson process with intensity function $D(x) = \lambda s(x)$ which is defined as follows:

$$f(X) \triangleq e^{-\lambda} \lambda^n s(x_1) \cdots s(x_n). \quad (2.86)$$

This is called *multi-dimensional Poisson distribution* [105] and an RFS having the distribution $f(X)$ is a multi-target Poisson process⁴.

The PHD filter is initialized by assuming $D_0(x)$ having the following form:

$$D_0(x) = n_0 s_0(x). \quad (2.87)$$

where $s_0(x)$ is a priori target distribution with n_0 initial expected number of targets. The form of $s_0(x)$ is determined by the specific implementation for the filter at hand, e.g. it can be a Gaussian mixture or a particle system approximation or even a *uniform PHD* if no a priori information is present.

Considering random target birth, spawning, disappearing, and Markov density for the temporal change in the multi-target distribution, the time prediction equation can be written as follows:

$$D_{k+1|k}(x) = b_{k+1|k}(x) + \int F_{k+1|k}(x|x') \cdot D_{k|k}(x') dx' \quad (2.88)$$

where

$$F_{k+1|k}(x|x') \triangleq p_s(x') f_{k+1|k}(x|x') + b_{k+1|k}(x|x').$$

is the so-called pseudo-Markov transition density encapsulating probability of target survival $p_s(x)$, spawned targets from existing ones $b_{k+1|k}(x|x')$, and single-target Markov density $f_{k+1|k}(x|x')$.

After the prediction step, the predicted expected number of targets $N_{k+1|k}$ is calculated as

$$N_{k+1|k} = \int D_{k+1|k}(x|Z^{(k)}) dx \quad (2.89)$$

$$= N_{k+1|k}^{birth} + N_{k+1|k}^{exist} + N_{k+1|k}^{spawned} \quad (2.90)$$

⁴This is a special case of a more general form point process called independent, identically distributed (i.i.d.) cluster process [86].

At time $k + 1$, a new measurement set $Z_{k+1} = \{z_1, \dots, z_m\}$ is obtained. With the predicted PHD $D_{k+1|k}(x|Z^{(k)})$, the PHD filter measurement update can be formulated as

$$D_{k+1|k+1}(x) \cong L_{Z_{k+1}}(x)D_{k+1|k}(x) \quad (2.91)$$

where $L_Z(x)$ is defined as follows:

$$L_Z(x) \triangleq 1 - p_D(x) + p_D(x) \sum_{z \in Z} \frac{L_z(x)}{\lambda c(z) + D_{k+1|k}[p_D L_z]}. \quad (2.92)$$

with $p_D(x)$ is the probability of detection of a target or sensor field of view, λ is the average number of clutter measurements with spatial distribution $c(z)$. After the measurement update, the posterior expected number of targets is

$$N_{k+1|k+1} = \int D_{k+1|k+1}(x) dx \quad (2.93)$$

$$= N_{k+1|k} - D_{k+1|k}[p_D] + \sum_{z \in Z_{k+1}} \frac{D_{k+1|k}[p_D L_z]}{\lambda c(x) + D_{k+1|k}[p_D L_z]}. \quad (2.94)$$

The PHD filter is one of the most fundamental multi-target filters in the RFS framework and has been extensively used since its first introduction to the literature. However, it only propagates the first-order moment of the posterior multi-target distribution. The information lost in the first-order moment approximation is mitigated by the so-called cardinalized PHD (CPHD) filter where the cardinality distribution $\rho_{k|k}(n)$ is propagated as well as the intensity function $D_{k|k}(x)$, at the expense of more computational complexity. In other words, the CPHD filter propagates the full probability distribution on the number of targets [92], [106]. It is a generalization of the PHD filter where the multi-target RFSs are modeled using i.i.d. cluster processes.

2.3.2.2. Generalized Labelled Multi-Bernoulli Filter

One of the most sophisticated multi-target trackers is the celebrated generalized multi-Bernoulli filter (GLMB). GLMB filter belongs to a class of filters where the target label is included in the target state and estimated along with the kinematic state [85]. Thus, it needs to calculate the explicit assignment of labels to target trajectory hypotheses in a mathematically rigorous manner.

In the GLMB filtering framework, targets are labeled by the ordered pair $\ell = (k, i)$, where i is the target index and k is the birth time of target i . The couple describes a target uniquely. An existing target at time k , therefore, has the state $(\mathbf{x}, \ell)$ where $\mathbf{x} \in \mathbb{X}$ and $\ell \in \mathbb{L}_{0:k}$. The *target label indicator* $\Delta(X) \triangleq \delta_{|X|}[|\mathcal{L}(X)|]$ is used to ensure that the labels, $\mathcal{L}(X)$, of the multi-target state X have the same cardinality. $\theta : \mathbb{L}_{0:k} \rightarrow \{0, 1, \dots, |Z|\}$ is the association map which assigns labels to measurements. The set of all possible association maps is denoted as Θ_k .

In the GLMB filter, the multi-target posterior density is of the following form [85]:

$$\pi_{k|k}(X|Z^k) = \Delta(X) \sum_{\xi \in \Theta_{0:k}} \omega_k^{(\xi)}(\mathcal{L}(X)) [p_k^{(\xi)}]^X \quad (2.95)$$

where $p_k^{(\xi)}(\cdot, \ell)$ is a probability density and $\omega_k^{(\xi)}(L)$ is a non-negative weight satisfying

$$\sum_{L \in \mathcal{F}(\mathbb{L})} \sum_{\xi \in \Theta_{0:k}} \omega_k^{(\xi)}(L) = 1.$$

The cardinality distribution of the GLMB density is defined as

$$\rho_k(n) = \sum_{L \in \mathcal{F}(\mathbb{L})} \sum_{\xi \in \Theta_{0:k}} \delta_n(|L|) \omega_k^{(\xi)}(L). \quad (2.96)$$

The GLMB density is a conjugate prior w.r.t. the standard multi-target measurement model (i.e. multi-target likelihood) and is closed under the multi-target prediction integral [85]. In other words, if the multi-target distribution is a GLMB at time k , the predicted and updated multi-target densities $\pi_{k+1|k}(\cdot)$ and $\pi_{k+1|k+1}(\cdot)$, respectively, are also GLMB densities. The following GLMB recursion equations constitute the first exact closed-form solution to the multi-target Bayes filter defined in (2.59)-(2.60). The multi-target predicted density is given by [85]

$$\pi_{k+1|k}(X|Z^k) = \Delta(X) \sum_{\xi \in \Theta_{0:k}} \omega_{k+1|k}^{(\xi)}(\mathcal{L}(X)) [p_{k+1|k}^{(\xi)}]^X \quad (2.97)$$

where

$$\omega_{k+1|k}^{(\xi)}(L) = \omega_{S,k+1|k}^{(\xi)}(L \cap \mathbb{L}_{0:k}) \omega_{\Gamma,k}(L \cap \mathbb{L}_{k+1}), \quad (2.98)$$

$$p_{k+1|k}^{(\xi)}(\mathbf{x}, \ell) = 1_{\mathbb{L}_{0:k}}(\ell) p_{S,k+1|k}^{(\xi)}(L \cap \mathbb{L}_{0:k}) + 1_{\mathbb{L}_k}(\ell) p_{\Gamma,k+1|k}^{(\xi)}(\mathbf{x}, \ell), \quad (2.99)$$

$$\omega_{S,k+1|k}^{(\xi)}(L) = [\bar{P}_{S,k+1|k}^{(\xi)}]^L \sum_{I \supseteq L} [1 - \bar{P}_{S,k+1|k}^{(\xi)}]^{I-L} \omega_k^{(\xi)}(I), \quad (2.100)$$

$$\bar{P}_{S,k+1|k}^{(\xi)}(\ell) = \left\langle P_{S,k+1|k}(\cdot, \ell), p_k^{(\xi)}(\cdot, \ell) \right\rangle, \quad (2.101)$$

$$p_{S,k+1|k}^{(\xi)}(\mathbf{x}, \ell) = \frac{\left\langle P_{S,k+1|k}(\cdot, \ell) f_{k+1|k}(\mathbf{x}, \cdot, \ell), p_k^{(\xi)}(\cdot, \ell) \right\rangle}{\bar{P}_{S,k+1|k}^{(\xi)}(\ell)}, \quad (2.102)$$

The measurement updated multi-target density $\pi_{k+1|k+1}(\cdot)$ is also a GLMB,

$$\pi_{k+1|k+1} = \Delta(X) \sum_{\xi \in \Theta_{0:k}} \sum_{\theta \in \Theta_{k+1}} \omega_{k+1}^{(\xi, \theta)}(\mathcal{L}(X)) [p_{k+1|k+1}^{(\xi, \theta)}(\cdot | Z_{k+1})]^X \quad (2.103)$$

where

$$\omega_{k+1}^{(\xi, \theta)}(L | Z) \propto 1_{\Theta_{k+1}(L)}(\theta) [\bar{\Psi}_{Z,k}^{(\xi, \theta)}]^L \omega_{k+1|k}^{(\xi)}(L), \quad (2.104)$$

$$\bar{\Psi}_{Z,k+1}^{(\xi, \theta)}(\ell) = \left\langle \Psi_{Z,k}^{(\xi, \theta)}(\cdot, \ell), p_{k+1|k}^{(\xi)}(\cdot, \ell) \right\rangle, \quad (2.105)$$

$$\Psi_{Z,k+1}^{(\xi, \theta)}(\mathbf{x}, \ell) = \begin{cases} \frac{\psi_{k+1}(z_{\theta(\ell)}; (\mathbf{x}, \ell))}{\lambda_{F,k}} & \text{if } \theta(\ell) > 0 \\ 1 - P_{D,k}(\mathbf{x}, \ell) & \text{if } \theta(\ell) = 0 \end{cases} \quad (2.106)$$

$$p_{k+1}^{(\xi, \theta)}(\mathbf{x}, \ell | Z) = \frac{\Psi_{Z,k+1}^{(\theta)}(\mathbf{x}, \ell) p_{k+1|k}^{(\xi)}(\mathbf{x}, \ell)}{\bar{\Psi}_{Z,k+1}^{(\xi, \theta)}(\ell)}. \quad (2.107)$$

The GLMB filter has been applied to many challenging problems from automotive systems to dense target tracking scenarios where the recent implementations can track a million targets [102]. A special case of the GLMB filter is called LMB where one-term approximation is assumed. The GLMB is considered a *true* multi-target tracker in the sense that the target tracks are propagated with unique labels. It is also an exact closed-form solution and therefore provably Bayes optimal [102].

2.3.2.3. Poisson Multi-Bernoulli Mixture Filter

Another example of the RFS-based multi-target tracking algorithms is the Poisson multi-Bernoulli mixture (PMBM) filter [103], [107]. The Poisson part

describes the so-called *undetected targets*, while the multi-Bernoulli part describes the distribution of the detected targets. Both the standard dynamic model and measurement model preserves the PMBM structure, therefore it is a conjugate prior under these model assumptions.

The PMBM model the multi-target distribution as a combination of a Poisson point process and a multi-Bernoulli mixture. The multi-target state, thus, is a union of two disjoint sets, X^u and X^d , where $X^u \cup X^d = X$, for undetected (up until current time) and detected (at least once) targets, respectively. While the Poisson component models the undetected targets, the MBM component models the detected targets.

Mathematically, the PMBM is expressed as follows [103].

$$f(X) = \sum_{X^u \uplus X^d} f^u(X^u) \sum_{j \in \mathbb{J}} \mathcal{W}^j f^j(X^d), \quad (2.108)$$

$$f^u(X^u) = e^{-\langle D^u, 1 \rangle} \prod_{x \in X^u} D^u(x), \quad (2.109)$$

$$f^j(X^d) = \sum_{\uplus_{i \in \mathbb{J}^j} X^i} \prod_{i \in \mathbb{J}^j} f^{j,i}(X^i). \quad (2.110)$$

where $f^{j,i}(\cdot)$ are Bernoulli components in the multi-Bernoulli mixture with $\mathcal{W}^j$ being the probability of the j th component. The PMBM is a conjugate prior which means the PMBM form is preserved through the prediction and update steps. Therefore, the filtering recursion reduces to computing the following parameter set, which completely defines a PMBM density.

$$D^u, \{(\mathcal{W}^j, \{r^{j,i}, f^{j,i}\}_{i \in \mathbb{J}^j})\}_{j \in \mathbb{J}}. \quad (2.111)$$

Given the multi-target posterior density as the PMBM, under the standard dynamic model, the predicted multi-target density (with the PMBM density parameter set) can be calculated as follows.

$$D_{k+1|k}^u(x) = D^b(x) + \langle D^u, p_S f_{k+1|k} \rangle, \quad (2.112)$$

$$r_{k+1|k}^{j,i} = \langle f^{j,i}, p_S \rangle r^{j,i}, \quad (2.113)$$

$$f_{k+1|k}^{j,i} = \frac{\langle f^{j,i}, p_S f_{k+1|k} \rangle}{\langle f^{j,i}, p_S \rangle}, \quad (2.114)$$

$$\mathcal{W}_{k+1|k}^j = \mathcal{W}^j. \quad (2.115)$$

where $D^b(x)$ is the intensity function of the birth process, p_S is the survival probability of the existing targets, $f_{k+1|k}(\cdot)$ is the single-target Markov density. Because of the conjugacy, the measurement updated multi-target distribution, under the standard measurement model, is also a PMBM density as well which can be written as follows.

$$f(X|Z) = \sum_{X^u \uplus X^d} f^u(X^u) \sum_{j \in \mathbb{J}_+} \sum_{A \in \mathcal{A}^j} \mathcal{W}_A^j f_A^j(X^d), \quad (2.116)$$

$$f^u(X^u) = e^{-\langle D^u, 1 \rangle} \prod_{x \in X^u} D^u(x), \quad (2.117)$$

$$f_A^j(X^d) = \sum_{\uplus_{C \in A} X^C} \prod_{C \in A} f_C^{j,i}(X^C). \quad (2.118)$$

$\mathbb{J}_+$ denotes the set of predicted multi-Bernoulli components and where the weights are

$$\mathcal{W}_A^j = \frac{\mathcal{W}_{k+1|k}^j \prod_{C \in A} \mathcal{L}_C}{\sum_{j' \in \mathbb{J}} \sum_{A' \in \mathcal{A}^{j'}} \mathcal{W}_{k+1|k}^{j'} \prod_{C' \in A'} \mathcal{L}_{C'}}, \quad (2.119)$$

$$\mathcal{L}_C = \begin{cases} \kappa^{\mathbf{C}_C} + \langle D_{k+1|k}^u, \ell_{\mathbf{C}_C} \rangle & \text{if } C \cap \mathbb{J} = \emptyset, |\mathbf{C}_C| = 1, \\ \langle D_{k+1|k}^u, \ell_{\mathbf{C}_C} \rangle & \text{if } C \cap \mathbb{J} = \emptyset, |\mathbf{C}_C| > 1, \\ 1 - r_{k+1|k}^{j,i_C} + r_{k+1|k}^{j,i_C} \langle f_{k+1|k}^{j,i_C}, q_D \rangle & \text{if } C \cap \mathbb{J} \neq \emptyset, |\mathbf{C}_C| = \emptyset, \\ r_{k+1|k}^{j,i_C} \langle f_{k+1|k}^{j,i_C}, \ell_{\mathbf{C}_C} \rangle & \text{if } C \cap \mathbb{J} \neq \emptyset, |\mathbf{C}_C| \neq \emptyset. \end{cases} \quad (2.120)$$

and the Bernoulli densities $f_C^j(X)$ are defined by the parameters

$$r_C^j = \begin{cases} \frac{\langle D_{k+1|k}^u, \ell_{\mathbf{C}_C} \rangle}{\kappa^{\mathbf{C}_C} + \langle D_{k+1|k}^u, \ell_{\mathbf{C}_C} \rangle} & \text{if } C \cap \mathbb{J} = \emptyset, |\mathbf{C}_C| = 1, \\ 1 & \text{if } C \cap \mathbb{J} = \emptyset, |\mathbf{C}_C| > 1, \\ \frac{r_{k+1|k}^{j,i_C} \langle f_{k+1|k}^{j,i_C}, q_D \rangle}{1 - r_{k+1|k}^{j,i_C} + r_{k+1|k}^{j,i_C} \langle f_{k+1|k}^{j,i_C}, q_D \rangle} & \text{if } C \cap \mathbb{J} \neq \emptyset, |\mathbf{C}_C| = \emptyset, \\ 1 & \text{if } C \cap \mathbb{J} \neq \emptyset, |\mathbf{C}_C| \neq \emptyset. \end{cases}, \quad (2.121)$$

$$f_C^j(X) = \begin{cases} \frac{\ell_{\mathbf{C}_C}(x) D_{k+1|k}^u(x)}{\langle D_{k+1|k}^u, \ell_{\mathbf{C}_C} \rangle} & \text{if } C \cap \mathbb{J} = \emptyset, \\ \frac{q_D(x) f_{k+1|k}^{j,i_C}(x)}{r_{k+1|k}^{j,i_C} \langle f_{k+1|k}^{j,i_C}, q_D \rangle} & \text{if } C \cap \mathbb{J} \neq \emptyset, \mathbf{C}_C = \emptyset, \\ \frac{\ell_{\mathbf{C}_C}(x) f_{k+1|k}^{j,i_C}(x)}{\langle f_{k+1|k}^{j,i_C}, \ell_{\mathbf{C}_C} \rangle} & \text{if } C \cap \mathbb{J} \neq \emptyset, \mathbf{C}_C \neq \emptyset. \end{cases} \quad (2.122)$$

The updated intensity function of the Poisson point process is $D^u(x) = q_D(x) D_{k+1|k}^u(x)$.

2.4. Multi-Sensor Fusion

Sensor fusion has become an important aspect of multi-sensor systems, mainly because with the decrease in the costs and size of the sensors and growing computational power, most systems depend on multiple and different sensor modalities to improve their reliability.

The use of multiple sensors in state estimation helps to reduce uncertainty and increase the reliability and robustness of the whole system under different conditions. While doing this, the aim is to use and combine all the available information efficiently, resulting in more consistent, accurate, and unified information than the individual sensors.

In multi-sensor state estimation, we do not know if the sources share common information [23]. Therefore, the so-called data incest is one of the most challenging issues in multi-sensor fusion. In this problem, the information is double-counted due to the network loops. Thus, it results that the estimates becoming biased. It is important to fuse the information from multiple data sources in a way that the common information from these sensors is removed. The multi-sensor information fusion can be done at different levels, regarding the data structure [5]. The complete data-fusion model is shown in Fig. 2.4.

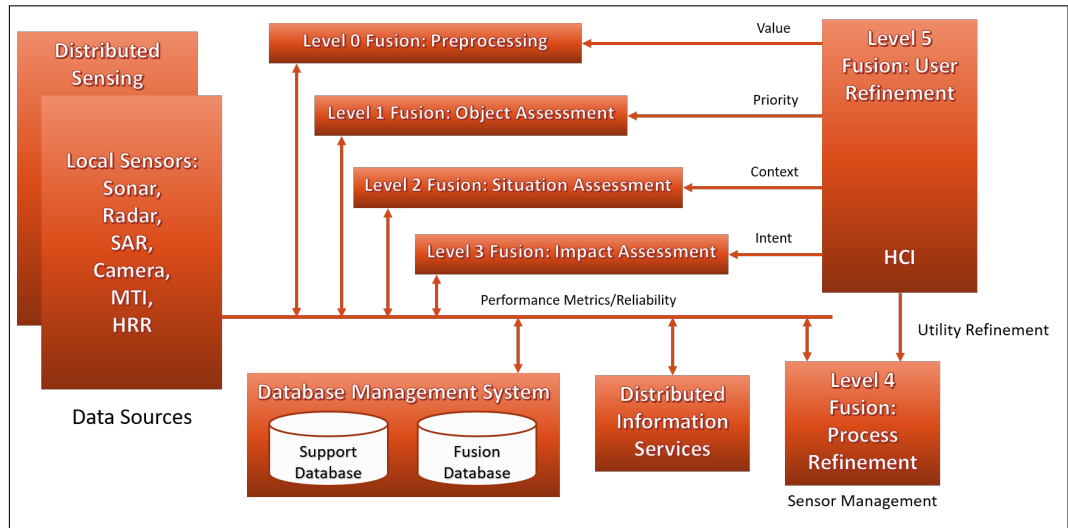


Figure 2.4: Complete data-fusion model and relation of each level with each other.

The most used sensor fusion schemes are measurement fusion and state fusion. In measurement fusion, the raw measurements from the individual sensors are transmitted

to a central node and fused together, and processed. This is also called centralized fusion. The overall system performance relies on the central processor to function properly. Therefore, its redundancy is decreased. In the state fusion scheme, each of the sensors processes its own measurements independently and generates its estimation of the environment. Then they transmit their estimations to a fusion node or share them with each other directly. Thus, fusing the individual estimations into a more reliable one. This is called decentralized/distributed sensor fusion. The centralized fusion produces the theoretically optimal solution if certain assumptions are satisfied, such as spatial and temporal data alignment. Thus, data synchronization is an important aspect of the sensor fusion scheme. In distributed fusion, however, each node/sensor processes its own measurements, thus a failure in one would not affect the others. Besides, the shared information would be in a reduced form rather than the raw sensor signal/measurement, thus the communication requirements are less than the centralized fusion.

2.4.1. Centralized Fusion

Centralized fusion is a data fusion approach used in target tracking, where data from multiple sensors or sources are combined to improve the accuracy and reliability of target position and trajectory estimates. In centralized fusion, all the sensor data is sent to a central processing unit or fusion center, which performs the data integration and target tracking tasks. Fig. 2.5 shows the general structure of the centralized fusion scheme.

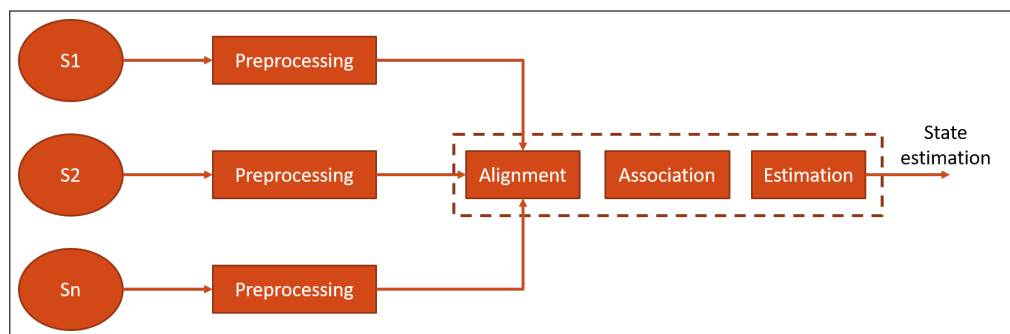


Figure 2.5: Centralized fusion architecture: measurement fusion.

For example, an overview of the centralized fusion scheme for target tracking is as follows:

- Data collection: Multiple sensors, such as radar, lidar, or cameras, collect raw data about the target's position, velocity, and other relevant attributes.
- Data transmission: The sensor data is transmitted to a central processing unit or fusion center, which can be a dedicated device or a software application running on a computer.
- Data preprocessing: The fusion center preprocesses the raw data, filtering out noise and converting it into a standard format that can be easily combined and analyzed.
- Data fusion: The fusion center combines the preprocessed data from all sensors using various fusion algorithms, such as Kalman filtering, particle filtering, or other statistical methods. The aim is to extract the most accurate and reliable information about the target's position, velocity, and trajectory.
- Target tracking: The fusion center continuously updates the target's estimated position and trajectory using the fused data, taking into account the target's motion dynamics and the uncertainties in the sensor measurements.
- Decision-making and control: Based on the fused target tracking information, decisions can be made regarding the tracking system's actions, such as adjusting the sensors' settings or directing the control of autonomous vehicles or weapons.

Centralized fusion in target tracking offers several advantages, including increased accuracy and reliability, as well as the ability to handle complex scenarios involving multiple targets and sensors. However, it also has some drawbacks, such as increased communication overhead, latency, and potential single points of failure due to the reliance on a central processing unit. In some cases, alternative approaches like distributed fusion or decentralized fusion may be more suitable, depending on the specific requirements of the target tracking system [6], [108].

Centralized fusion usually includes a Kalman (or a similar appropriate) filter to optimally fuse the individual measurements into a more reliable estimate. It is also called *measurement fusion* since individual estimates (from raw signals) are low-level sensor measurements. This scheme is considered to be able to produce *optimal* estimates, compared to the distributed fusion scheme, albeit at a communication overhead. Also, reliance on a central processing unit creates a single point of failure, making the system potentially more vulnerable to faults or attacks. The centralized fusion offers a more straightforward and easier-to-implement approach to data fusion but can face challenges related to communication overhead, scalability, and robustness.

2.4.2. Distributed Fusion

Contrary to the centralized architecture, distributed data fusion (DDF), mainly, provides scalability and robustness to the overall system. It is possible to use different types of sensors in a plug-and-play sense. Losing any of the subsystems in a distributed architecture does not prevent the functioning of the whole system, because individual nodes continue to process their own measurement. Similarly, to add a new system to the network, simply a link between the nodes can be established. The general flow of the DDF can be seen in Fig. 2.6.

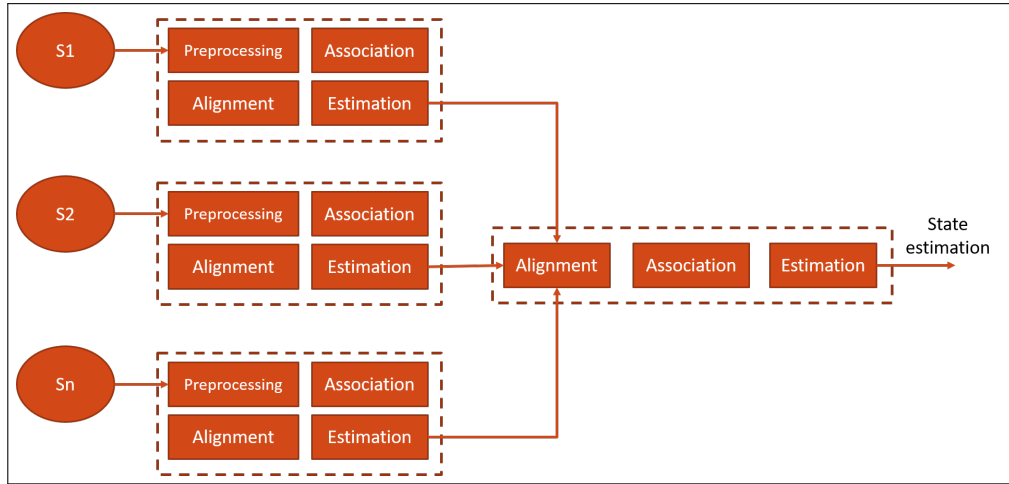


Figure 2.6: Distributed fusion architecture: state fusion.

Despite these advantages, because of the information flow and feedback loops, information may be used repetitively in the nodes. When this information is fused blindly, the fused estimate becomes corrupted. Thus, this is a fundamental theoretical problem in the DDF framework, that can not be overcome using general Kalman filtering [108].

For example, to estimate a vehicle's position in a distributed network, acceleration estimates from wheel speed, laser gyros and pressure sensors on the accelerator pedal can be fused. In this concept, each sensing node shares its estimate as the mean and covariance information to be fused together, and a better estimate can be obtained [108]. However, using a Kalman filtering framework for this purpose generally results in an under-estimated fused covariance because of the temporal correlations. The individual estimates must be independent or their correlations must be known or calculated and factored out. Otherwise, the fused estimate will

be corrupted by so-called double-counted information. Calculating and maintaining cross-covariance between different nodes is not feasible in most network topologies. Thus, to achieve robust and consistent estimates in a DDF architecture is to bypass the inter-node independence assumptions.

For a theoretically rigorous and unified fusion approach, first, some definitions should be made. Let us be fusing information from two sources in the form of probability distributions. For this, we need abstract addition and multiplication rules as follows [58],[109]:

$$(p_1 \oplus p_2)(X) = \frac{p_1(X)p_2(X)}{\int p_1(X')p_2(X')\delta X'} \quad (2.123)$$

$$(\omega_1 \otimes p_1)(X) = \frac{p_1(X)^\omega}{\int p_1(X')\delta X'} \quad (2.124)$$

where $f(\cdot)$ is a probability distribution and ω is a scalar fusion weight. In the above definitions, $\oplus$ should be commutative and associative, and $\otimes$ should be distributive w.r.t. $\oplus$. These operators enable recursive fusion of the form

$$p_{1,\dots,s} = (\omega_1 \otimes p_1) \oplus \dots \oplus (\omega_s \otimes p_s) \quad (2.125)$$

The solution in (2.125) is optimal in the limit case and also is immune to double-counting [110].

2.4.2.1. Exponential Mixture Fusion (Geometric Average)

A specific form of the above abstraction is the exponential mixture fusion⁵ [111], [112] is as follows.

$$p_{CF}(x) = \frac{p_1^\omega(x)p_2^{1-\omega}(x)}{\int p_1^\omega(x)p_2^{1-\omega}(x)dx} \quad (2.126)$$

where $\omega \in [0, 1]$ is usually found to be the solution to the following optimization problem. On a side note, (2.126) is a special case of *logarithmic opinion pooling* when only two sources are being pooled [6].

⁵In literature, this is also referred to as geometric average, generalized covariance intersection, or Chernoff fusion rule

$$\omega^* = \arg \min_{\omega \in [0,1]} \mathcal{L}(\omega) \quad (2.127)$$

where $\mathcal{L}(\cdot)$ is a suitable objective function. The relation between (2.123) and (2.126) is apparent. As stated in [61], the Chernoff fusion is immune to data-incest (or double counting). In the case of $p_1(\cdot)$ and $p_2(\cdot)$ being Gaussian densities, the expression in (2.126) admits a closed form solution which is widely known as covariance intersection (CI) or geometric average (GA)⁶:

$$P_{GA}^{-1}x_{GA} = \omega P_1^{-1}x_1 + (1 - \omega)P_2^{-1}x_2 \quad (2.128)$$

$$P_{GA}^{-1} = \omega P_1^{-1} + (1 - \omega)P_2^{-1} \quad (2.129)$$

where x_i and P_i are the mean and covariance matrix of each Gaussian, respectively. There is a geometrical interpretation of (2.128-2.129): When the covariance ellipses for P_1 , P_2 and P_{GA} are plotted, P_{GA} will always lie within the intersection of individual covariances [108], hence the term *covariance intersection*. This is for all the possible cross-covariances, hence it is immune to double-counting⁷. The GA algorithm is a convex combination of the individual covariances as a function of free parameter ω .

The *GA* fusion makes no assumptions about the independence of the fusing estimates and is suitable for arbitrary fusion networks. Hence, it provides an important and theoretically rigorous way to tackle the data fusion problem in a distributed fusion manner.

2.4.2.2. Arithmetic Average Fusion

Arithmetic average (*AA*) fusion is a method used in data fusion, which is the process of combining multiple data sources to produce more accurate, reliable, and useful information than can be obtained from a single source. In the context of data fusion, arithmetic average fusion is a simple technique for combining data points or values from different sources by calculating the average of these values.

AA fusion is easy to implement and understand, making it a popular choice for simple data fusion tasks. However, it may not always be the best choice, especially in cases where the data sources have varying reliability, or when more advanced fusion techniques are necessary to capture the nuances of the data. In such cases, other

⁶In the remainder of this dissertation, we will refer to the more general exponential mixture fusion as the *GA* for contextual coherence.

⁷The complete illustration can be found in [108]

methods such as weighted averages or more sophisticated fusion algorithms might be more suitable.

AA fusion is related to the linear opinion pooling [113]. While *GA* fusion has been more widely studied in the literature, *AA* fusion is relatively new. Also, the *AA* fusion shares some common properties with the *GA* fusion such as being immune to double counting and more conservative than the counterpart. It also enjoys efficient computational complexity, compared to *GA* fusion. However, it trade-offs these advantages at the price of an inflated covariance estimate. The support of the fused pdf is the union of the individual pdfs.

In terms of fusing distributions (i.e. sources), the *AA* fusion is defined by the following expression.

$$p_{AA}(x) = \sum_{i \in \mathcal{I}} \omega_i p_i(x). \quad (2.130)$$

If the separate estimations are unbiased, the Arithmetic Average (*AA*) fusion maintains the unbiasedness, implying that the fused estimation is also unbiased. This stands in contrast to the Geometric Average (*GA*) fusion, as discussed in [113]. The variance of the *AA* fusion has an upper bound equal to the largest variance among the individual fusing estimations. The lower bound of its variance is less than the smallest variance of the fusing estimations, provided that the correlation coefficient between any two estimations adheres to a specific identity:

$$\rho < \left(\frac{\min(P_1, P_2)}{\max(P_1, P_2)} \right)^2 \quad (2.131)$$

otherwise, the lower bound is determined by the minimum variance among the individual fusing estimations, where the P_1 and P_2 are the covariances of individual estimates.

2.5. Performance Metrics

2.5.1. Multi-Object Miss Distance

In single-sensor single-target tracking, the most common approach to assess the performance of a filter is to use the Mahalanobis distance between the state estimate $\hat{x}_k$ and the ground truth x_k , where the quantities are Euclidean vectors. The Mahalanobis

distance $d(x_k, \hat{x}_k)$, between two Euclidean vectors is

$$d(x, \hat{x}_k) = \sqrt{(x - \hat{x}_k)^T C^{-1} (x - \hat{x}_k)}. \quad (2.132)$$

In multi-sensor multi-target tracking, however, the sets of target truths (or ground truth) $X_k = \{x_{k,1}, \dots, x_{k,n_k}\}$ and target estimates $\hat{X}_k = \{\hat{x}_{k,1}, \dots, \hat{x}_{k,\hat{n}_k}\}$ are modeled as random finite sets. Thus, the *distance* concept must be extended to random sets X_k and $\hat{X}_k$. This section summarizes the widely used multi-sensor multi-target performance (distance) metrics in the literature. In the mathematical sense, a metric must have the following properties:

- Symmetry: $d(X, X') = d(X'X)$.
- Non-negativity: $d(X, X') \geq 0$.
- Definiteness: $d(X, X') = 0$ iff $X = X'$.
- Triangle Inequality: $d(X, X') \leq d(X, X'') + d(X', X'')$ for any X, X', X'' .

2.5.1.1. Hausdorff Distance

The most well-known distance metric for subsets is the Hausdorff distance[33] which is defined by

$$d^H(X, X') = \begin{cases} \infty & \text{if } X = \emptyset \text{ or } X' = \emptyset \\ \max \{ \max_{\mathbf{x} \in X} \min_{\mathbf{x}' \in X'} d(\mathbf{x}, \mathbf{x}'), \max_{\mathbf{x}' \in X'} \min_{\mathbf{x} \in X} d(\mathbf{x}, \mathbf{x}') \} & \text{if o.w.} \end{cases} \quad (2.133)$$

where $d(\mathbf{x}, \mathbf{x}')$ is some metric on $\mathfrak{Y}$. While the Hausdorff metric is consistent with FISST, it is not so much adequate as a performance metric for multi-target tracking problems. The reason is that it is insensitive to cardinality differences and outliers.

2.5.1.2. Wasserstein Distance

The following miss distance:

$$d(X, X') = \min_{\pi} \sqrt{\frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{x}'_{\pi, i}\|^2} \quad (2.134)$$

where the minimum operator is over all permutations π on $1, \dots, n$ describes the smallest RMS error between the elements of two sets with the same cardinality, $n = n'$. Mahler generalized it to the case $n \neq n'$, as a true metric, which is called the Wasserstein distance:

$$d_p^W(X, X') \triangleq \inf_C \sqrt[p]{\sum_{i=1}^n \sum_{i'=1}^{n'} C_{i,i'} d(\mathbf{x}_i, \mathbf{x}'_{i'})^p}. \quad (2.135)$$

where the *infimum* is over $n \times n'$ *transportation matrices* C [33]. A matrix is called a transportation matrix if $C_{i,i'} \geq 0$ and

$$\sum_{i=1}^n C_{i,i'} = \frac{1}{n'}, \sum_{i'=1}^{n'} C_{i,i'} = \frac{1}{n}. \quad (2.136)$$

However, the Wasserstein distance is found to be suffering from non-intuitive behaviors which are detailed in [33].

2.5.1.3. Optimal Subpattern Assignment (OSPA) metric

In [114] a multi-object distance metric is proposed which is mathematically intuitive called *optimum subpattern assignment* (OSPA) metric⁸. This metric uses the following:

- a baseline metric (i.e. single-object metric) $d(\mathbf{x}, \mathbf{x}')$ between single-target states $\mathbf{x}$ and $\mathbf{x}'$.
- a real number $c > 0$ which has the same unit of measurement as $\mathbf{x}$ and performs as a cutoff.
- a real number $p \geq 1$ which is unitless.

Assume that the estimated track set is denoted as $X = \{x_1, \dots, x_n\}$ and the set of truths is denoted as $Y = \{y_1, \dots, y_m\}$. Further assume that, wlog, $0 < n \leq m$. The OSPA distance between X and Y is defined as

⁸Readers can refer to [33], [114] for more detail on multi-object performance metrics

$$d_{p,c}^{OSPA}(X, Y) = \left(\min_{\pi} \frac{1}{m} \sum_{i=1}^n d_c(x_i, y_{\pi,i})^p + \frac{c^p}{m} (m - n) \right)^{1/p}. \quad (2.137)$$

where the minimum is over all permutations π on $1, \dots, m$ and $d_c(\cdot, \cdot) = \min\{c, d(\cdot, \cdot)\}$. If $0 = n \leq m$, then, by convention, $d_{p,c}^{OSPA}(\emptyset, Y) = c$.

The parameter c emphasizes the importance of the cardinality difference as opposed to the localization error. The parameter p determines the sensitivity to statistical outliers. The first minimizing term in (2.137) corresponds to the localization accuracy w.r.t. the truth. The base metric measures the error in the estimated position and can be any single-target error metric. The second part in (2.137) measures the error in cardinality estimation. In other words, the error in the estimated number of objects is reflected in the total estimation error, by using c . Note that, it is assumed $n \leq m$, meaning that there is no falsely detected/estimated target track but there can be missed target tracks. Intuitively, the reverse can also be considered since the same as the metric is a symmetric function.

In the plain form as given in (2.137), the OSPA metric measures the instantaneous distance between two sets. In general, however, there should be a way to address the temporal aspect of the tracking process. In other words, to appropriately determine the performance of a multi-target tracker, the error metric must be defined over an extended period of time [33]. Additionally, for more general *multi-target track (or estimation) fusion algorithms*, there is also the track association problem to be considered when evaluating the performance of the fusion algorithm. Therefore, the appropriate extensions of the OSPA metric should be taken into consideration for performance evaluation [115].

2.5.2. Multi-Object Information Functionals

The single-object miss distance metrics measure the dissimilarity between two states. A more general approach is to measure the dissimilarity (or similarity) between *probability distributions* that are functions of states. The concept is usually referred to as the *information-theoretic* functional [33]. Familiar methods such as Kullback-Leibler, Csiszár functional, or Cauchy-Schwarz divergence are examples of these information-theoretic functionals.

Through a unitless, non-negative convex kernel $c(x)$ of variable $x \geq 0$ and

$c(1) = 0$, the discrimination between two multi-object probability functions $f_1(Y)$ and $f_0(Y)$ can be calculated as follows:

$$I_c(f_1 : f_0) = \int c \left(\frac{f_1(Y)}{f_0(Y)} \right) f_0(Y) \delta Y. \quad (2.138)$$

The above expression gives the Csiszár information functional associated with $c(x)$. Another instance of information functional is the Kullback-Leibler divergence which is found when $c(x) = 1 - x + x \log x$:

$$I_c(f_1 : f_0) = \int \log \left(\frac{f_1(Y)}{f_0(Y)} \right) f_1(Y) \delta Y. \quad (2.139)$$

Other possible selections for $c(x)$ are as follows [33]:

- Chi-squared- $c(x) = (x - 1)^2$:

$$I_c(f_1 : f_0) = -1 + \int \frac{f_1(Y)^2}{f_0(Y)} \delta Y. \quad (2.140)$$

- L^1 metric- $c(x) = |x - 1|$:

$$I_c(f_1 : f_0) = \int |f_1(Y) - f_0(Y)| \delta Y. \quad (2.141)$$

- Hellinger- $(\sqrt{x} - 1)^2$:

$$I_c(f_1 : f_0) = 2 - 2 \int \sqrt{f_1(Y) f_0(Y)} \delta Y. \quad (2.142)$$

2.6. Integrated Navigation

Navigation is, basically, finding and/or specifying a route from one point in space to another. Whether the medium is sea, land, air, or even deep space, to be able to travel from one location to a destination it is crucial to know where one is and where they are headed, at least with respect to known objects or landmarks.

Since ancient times, accurate navigation skills have been crucial for explorers and adventurers. Ancient Greeks mostly used the stars and celestial objects to navigate the Mediterranean. It is written in Homer's *Odyssey* that Odysseus is told by Calypso, a river nymph, to keep the Ursa Major on his left and to observe The Seven Sisters' (another constellation) position, as he sails from Ogygia, a Mediterranean island, to east.

As time progressed and technology advanced, many methods and devices have been developed and used to navigate the globe safely, such as a magnetic compass, astrolabe, or even an Icelandic sunstone (by the medieval Vikings). However, modern navigation techniques are much more complex, principled, and naturally accurate.

The inertial navigation concept, which depends on the use of linear and rotational acceleration measurements and the use of Newtonian mechanics, provides very accurate positioning at very high rates, during short time intervals. By integrating these inertial measurements, measured by accelerometers and gyroscopes, generally strapped to the platform rigidly, position and rotational motion can be evaluated in time. Then, an external aiding system is used to estimate the inertial sensor errors and bound the time-dependent errors. This operation is called integrated navigation. A generic architecture for integrated navigation systems is shown in Fig. 2.7.

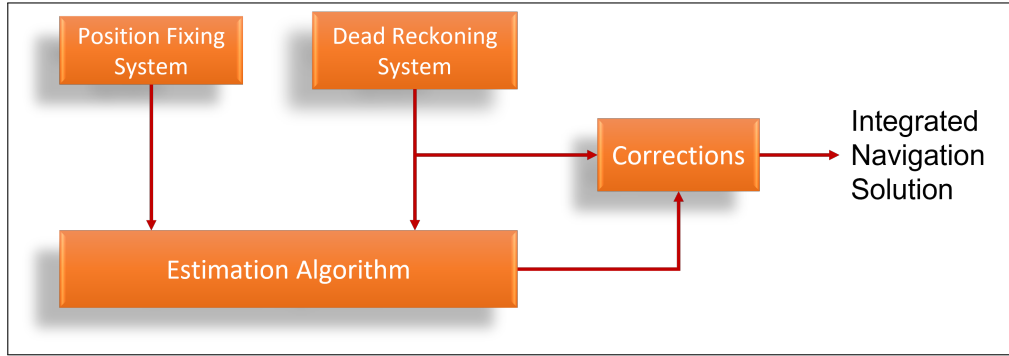


Figure 2.7: A generic integrated navigation architecture.

2.6.1. Integrated (Strapdown) Navigation Equations

In this section, the mathematical formulation of the integrated navigation using inertial sensor data is presented without much detail. Also, by mathematically modeling the errors on the navigation states, it is shown how these variables change and how the errors propagate over time.

The block diagram of a strapdown navigation system is shown in Fig. 2.8 (taken from [116]). Linear and angular motions in each axis are sensed by the accelerometers and gyroscopes which are mounted onto the vehicle rigidly. The rotation rate of the body, $\omega_{i/b}^b$ is measured w.r.t the body (frame), while body-referenced accelerometer measurements, f^b are transformed to the (local) navigation frame using

a transformation matrix C_b^n (or an equivalent mathematical tool, such as quaternion) [116]. The matrix is evaluated by using the measurements from the gyros and navigation frame rate which is due to the vehicle's movement.

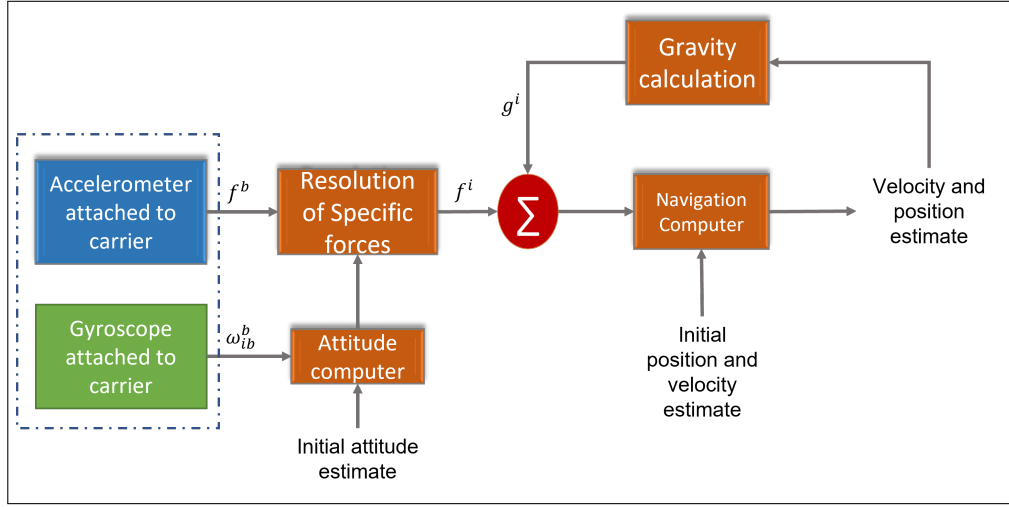


Figure 2.8: Strapdown navigation mechanization.

2.6.1.1. Velocity Equation

The velocity vector in the local-level frame (n), relative to the Earth, is defined in terms of ECEF (Earth-centered, Earth-fixed) coordinates (e) as follows

$$v^n = C_e^n \dot{r}^e. \quad (2.143)$$

Then, if we take the derivative of (2.143) w.r.t. time,

$$\dot{v}^n = C_e^n \ddot{r}^e + \dot{C}_e^n \dot{r}^e.$$

The objective here is to obtain a result that relates the specific force measured by the accelerometer to the rate of change in velocity.

The ECEF position vector in a non-rotating inertial frame is defined as

$$r^e = C_i^e r^i.$$

and its time derivative is

$$\dot{r}^e = C_i^e \dot{r}^i + \dot{C}_i^e r^i.$$

The time derivative of the direction cosine matrix C_i^e is

$$\dot{C}_i^e = -C_i^e \Omega_{i/e}^i.$$

then, if we substitute into the previous equation we get

$$\dot{r}^e = C_i^e (\dot{r}^i - \Omega_{i/e}^i r^i). \quad (2.144)$$

By taking the 2nd time derivative of (2.144) and rearranging the terms we get

$$\ddot{r}^e = C_i^e (\ddot{r}^i - 2\Omega_{i/e}^i \dot{r}^i + \Omega_{i/e}^i \Omega_{i/e}^i r^i). \quad (2.145)$$

The final velocity equation evolution in the navigation frame⁹ is obtained after applying specific force and gravitational acceleration as

$$\dot{v}^n = f^n - (\Omega_{e/n}^n + 2\Omega_{i/e}^n) v^n + g^n. \quad (2.146)$$

2.6.1.2. Attitude Equations

The attitude dynamics of the INS are represented as a direction cosine matrix in a differential equation form. The time evolution of the dynamics of attitude representation is obtained as follows

$$\dot{C}_b^n = -\Omega_{b/n}^n C_b^n$$

The matrix $\Omega_{b/n}^n$ is formed from the rotation vector $\omega_{b/n}^n$ (by skew-symmetrization) which is obtained from the angular velocity theorem [116]

$$\omega_{b/n}^n = \omega_{i/n}^n - C_b^n \omega_{i/b}^b \quad (2.147)$$

The vector $\omega_{i/b}^b$ is the output measurement of the strapdown gyros.

⁹The equations are described in the local level navigation frame. The reader should note that other reference frames, such as ECEF, local-level geodetic, or wander-azimuth formulations exist. The specification of the frame is necessary when writing the components and defining the rotation matrix elements.

2.6.1.3. Position Equations

The position equation is represented as the differential equation in the following

$$\dot{C}_e^n = -\Omega_{e/n}^n C_e^n \quad (2.148)$$

2.6.2. Inertial Navigation System Error Model

In this section, the navigation system errors are modeled in the form of nonlinear differential equations. These equations describe the behavior of the navigation states in time and are used as the process model in the following chapters.

2.6.2.1. Velocity Error Equations

The velocity equation obtained in (2.146) can be written in the vector form as follows

$$\dot{v}^n = f^n - (\omega_{e/n}^n + 2\omega_{i/e}^n) \times v^n + g^n \quad (2.149)$$

The computed velocity is erroneous due to imperfections in initialization, inertial sensor instruments, etc. The resulting erroneous velocity equation can be written as

$$\dot{\bar{v}}^n = \bar{f}^n - (\bar{\omega}_{e/n}^n + 2\bar{\omega}_{i/e}^n) \times \bar{v}^n + \bar{g}^n \quad (2.150)$$

The relation between the true velocity in (2.149) and the perturbed velocity in (2.150) is

$$\bar{v}^n = v^n + \delta v^n \quad (2.151)$$

The same approach can be applied to other terms:

$$\bar{f}^n = \bar{C}_b^n \bar{f}^b = [I - (\phi \times)] C_b^n (f^b + \delta f^b) \approx [I - (\phi \times)] f^n + \delta f^n \quad (2.152)$$

$$\bar{\omega}_{e/n}^n = \omega_{e/n}^n + \delta \omega_{e/n}^n \quad (2.153)$$

$$\bar{\omega}_{i/e}^n = \omega_{i/e}^n + \delta \omega_{i/e}^n \quad (2.154)$$

$$\bar{g}^n = g^n + \delta g^n \quad (2.155)$$

Then, by substituting these quantities into (2.150) and rearranging terms we get the dynamic equation of the "velocity error" in the navigation frame.

$$\begin{aligned} \delta \dot{v}^n = & -(\delta \omega_{e/n}^n + 2\delta \omega_{i/e}^n) \times v^n - (\omega_{e/n}^n + 2\omega_{i/e}^n) \times \delta v^n \\ & + f^n \times \phi + \delta f^n + \delta g^n \end{aligned} \quad (2.156)$$

The "error rotation vectors" $\delta \omega_{e/n}^n$ and $\delta \omega_{i/e}^n$ in (2.156) are functions of velocity and position perturbations, respectively [116]. The rotation vectors and their error form expressions are as follows

$$\omega_{e/n}^n \approx \begin{bmatrix} \frac{-v_y^n}{R} \\ \frac{v_x^n}{R} \\ \rho_z \end{bmatrix} \quad (2.157)$$

and

$$\delta \omega_{e/n}^n \approx \begin{bmatrix} \frac{-\delta v_y^n}{R} - \frac{\rho_x}{R} \delta h \\ \frac{\delta v_x^n}{R} - \frac{\rho_y}{R} \delta h \\ \delta \rho_z \end{bmatrix} \quad (2.158)$$

where R is the approximated nominal Earth radius, $\delta R \approx \delta h$ and ρ is called transport rate. The error in Earth rotation vector $\omega_{i/e}^n$ is

$$\delta \omega_{i/e}^n = \omega_{i/e}^n \times \delta \theta \quad (2.159)$$

2.6.2.2. Attitude Error Equations

The attitude errors can be described from the body to the navigation frame in different formulations. The formulations in this section are called ϕ or tilt error formulation [116]. The erroneous attitude transformation matrix C_b^n is

$$\bar{C}_b^n = [I - (\phi \times)] C_b^n \quad (2.160)$$

Then, the error in the direction cosine matrix is

$$\begin{aligned} \delta C_b^n &= \bar{C}_b^n - C_b^n \\ &= (\phi \times) C_b^n \end{aligned}$$

By taking the derivative of both lines and solving for $\dot{\phi}$ we get

$$\delta \dot{C}_b^n = -(\dot{\phi} \times) C_b^n - (\phi \times) \dot{C}_b^n$$

where the derivative of the body to navigation transform is defined as $\dot{C}_b^n = -\Omega_{b/n}^n C_b^n$. Then by extending the terms, it yields

$$(\dot{\phi} \times) = -\Omega_{b/n}^n (\phi \times) + (\phi \times) \Omega_{b/n}^n + (\bar{\Omega}_{b/n}^n - \Omega_{b/n}^n)$$

or in the vector form

$$\dot{\phi} = \phi \times \omega_{b/n}^n + (\bar{\omega}_{b/n}^n - \omega_{b/n}^n) \quad (2.161)$$

After expanding the angular rate terms¹⁰, we obtain the attitude error equation as in

$$\dot{\phi} = \delta \rho + \omega_{i/e}^n \times \delta \theta + \phi \times \omega_{i/n}^n + \varepsilon^n \quad (2.162)$$

2.6.2.3. Position Error Equations

The position error formula is obtained by a similar approach to the velocity and attitude equations. The inclined reader is referred to the original text reference [116] for the detailed derivations. The matrix and vector form of the position error equation, respectively, are

$$(\delta \dot{\theta} \times) = (\delta \rho \times) - \Omega_{e/n}^n (\delta \theta \times) + (\delta \theta \times) \Omega_{e/n}^n \quad (2.163)$$

$$\delta \dot{\theta} = \delta \rho - \omega_{e/n}^n \times \delta \theta \quad (2.164)$$

In this section, the system error dynamics are derived in terms of differential equations. These equations when brought together in the form of a system of equations,

¹⁰The expressions for detailed derivation is present in several references such as [116].

describe the time evolution of the errors in the navigation state variables. In the matrix form, the error equations can be written as

$$\frac{d}{dt}\delta x(t) = F(t)\delta x(t) + G(t)w_s(t) \quad (2.165)$$

where $[\delta\theta_x, \delta\theta_y, \delta h, \delta v_x^n, \delta v_y^n, \delta v_z^n, \phi_x, \phi_y, \phi_z, \delta\theta_z]^T$ are the components of error state vector. The discrete-time state transition matrix can be linearized (as in conventional Kalman filtering) and obtained by

$$\Phi_k = \exp(F_k \tau_s) \quad (2.166)$$

This is the generic methodology where the nonlinear system error propagation can be used to model the time update step of the Kalman filter¹¹.

2.7. Appendices

2.7.1. Multi-Object Probability Distributions

A multi-object density is a real-valued function $f(Y)$ defined on $\mathfrak{Z}_0$ where the finite-set variable is a subset $Y \subseteq \mathfrak{Z}_0$. If $\mathfrak{Z}_0$ has the unit of measurement K , then $f(Y)$ has the unit of measurement $K^{-|Y|}$ where $|Y|$ is the number of elements (or cardinality) in finite set Y . The set integral of multi-object densities is linear:

$$\int_S (a_1 f_1(Y) + a_2 f_2(Y)) \delta Y = a_1 \int_S f_1(Y) \delta Y + a_2 \int_S f_2(Y) \delta Y. \quad (2.167)$$

However, they are not additive in S (including the case $S_1 \cap S_2 = \emptyset$):

$$\int_{S_1 \cup S_2} f(Y) \delta Y \neq \int_{S_1} f(Y) \delta Y + \int_{S_2} f(Y) \delta Y. \quad (2.168)$$

A multi-object density is, as its single-object equivalent, a multi-object probability density function as long as it satisfies the following:

¹¹We have omitted the elements of the matrix F which is a 15×15 -matrix, in most applications. For details, the reader can refer to relevant references [116], [117]

$$\int f(Y) \delta Y = 1 \quad (2.169)$$

$$f(Y) \geq 0. \quad (2.170)$$

The probability density function $f_\Psi(Y)$ of an RFS Ψ is

$$\int_S f_\Psi(Y) \delta Y = \Pr(\Psi \subseteq S) \quad (2.171)$$

for all S . Similarly, the following definitions are used throughout the text:

$$\int_{\Psi \subseteq S} f_\Psi(Y) \delta Y \triangleq \Pr(\Psi \subseteq S) \quad (2.172)$$

$$= \int_S f_\Psi(Y) \delta Y \quad (2.173)$$

$$\int_{|\Psi|=n} f_\Psi(Y) \delta Y \triangleq \Pr(|\Psi| = n) \quad (2.174)$$

$$= \frac{1}{n!} \int f_\Psi(\{\mathbf{y}_1, \dots, \mathbf{y}_n\}) d\mathbf{y}_1 \dots d\mathbf{y}_n \quad (2.175)$$

$$\int_{\Psi \ni \mathbf{y}} f_\Psi(Y) \delta Y \triangleq \Pr(\mathbf{y} \in \Psi) \quad (2.176)$$

$$= \int f_\Psi(\{\mathbf{y}\} \cup W) \delta W. \quad (2.177)$$

The cardinality distribution of a random finite set Ψ is defined as follows:

$$p_\Psi(n) \triangleq p_{|\Psi|}(n) \quad (2.178)$$

$$\triangleq \Pr(|\Psi| = n) \quad (2.179)$$

$$= \frac{1}{n!} \int f_\Psi(\{\mathbf{y}_1, \dots, \mathbf{y}_n\}) d\mathbf{y}_1 \dots d\mathbf{y}_n. \quad (2.180)$$

The $n!$ term is due to the reason that the probability assigned to a finite set must be equally distributed among all the permutations of the variables in that set [33].

2.7.2. Important Multi-Object RFSs

There are several multi-object probability distributions that are widely used in developing multi-sensor multi-target tracking algorithms. In the following, these multi-object distributions are introduced. For more detail on these see [33].

2.7.2.1. Independent, Identically Distributed (i.i.d.) Cluster Process

Suppose that $p(n)$ is a probability distribution on non-negative integers. Also, let $f(\mathbf{y})$ be a probability distribution on $\mathfrak{Y}_0$. For any finite-set $Y = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$ with $|Y| = n$, the density $f(Y)$ is defined as

$$f(Y) \triangleq n!p(n)f(\mathbf{y}_1) \cdots f(\mathbf{y}_n). \quad (2.181)$$

It can be shown that the above density satisfies $\int f(Y)\delta Y = 1$, so it is a probability density.

2.7.2.2. Multi-Object Poisson Process

If the cardinality distribution $p(n) = e^{-\lambda}\lambda^n/n!$ which is a Poisson distribution, then the multi-object distribution becomes

$$f(Y) \triangleq e^{-\lambda}\lambda^n f(\mathbf{y}_1) \cdots f(\mathbf{y}_n). \quad (2.182)$$

which is called the *multidimensional Poisson distribution* [86]. Thus, a multi-object distribution of this form is called a Poisson (point) process. The following definition

$$I(\mathbf{y}) \triangleq \lambda \cdot f(\mathbf{y}) \quad (2.183)$$

is called the intensity function of the Poisson point process.

2.7.2.3. Multi-Object Dirac Delta Density

Another general distribution is the Dirac delta density which is defined as follows:

$$\delta_{Y'}(Y) \triangleq \begin{cases} 0 & \text{if } |Y| \neq |Y'| \\ 1 & \text{if } Y = Y' = \emptyset \\ \Sigma_{\sigma} \delta_{\mathbf{y}'_{\sigma_1}}(\mathbf{y}_1) \cdots \delta_{\mathbf{y}'_{\sigma_n}}(\mathbf{y}_n) & \text{if } \begin{cases} Y = \{\mathbf{y}_1, \dots, \mathbf{y}_n\} \\ Y' = \{\mathbf{y}'_1, \dots, \mathbf{y}'_n\} \end{cases} \end{cases} \quad (2.184)$$

where the σ denotes all the permutations on $1, \dots, n$ over which the summation is taken. $\delta_{Y'}(Y)$ satisfies the following property:

$$\int \delta_{Y'}(Y) \cdots f(Y) \delta Y = f(Y') \quad (2.185)$$

for a finite set Y .

2.7.2.4. Multi-Object Uniform Process

Let $\mathfrak{Y}_0 = \mathbb{R}^N$ and suppose that there are no more than $\check{n}$ objects in a domain $D \subseteq \mathfrak{Y}_0$ of hyper-volume $|D|$. A multi-object uniform distribution of $Y = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$ is defined as

$$u_{D, \check{n}(Y)} = \frac{n!}{|D|^n \cdot (\check{n} + 1)} \quad (2.186)$$

if $Y \subseteq D$. The similarity between the standard uniform distribution and the multi-object uniform process can be easily seen.

2.7.2.5. Multi-Bernoulli Process

Suppose that $f_1(\mathbf{y}), \dots, f_\nu(\mathbf{y})$ are probability distributions on $\mathfrak{Y}_0$ and $0 \leq q_1, \dots, q_\nu < 1$. Then the multi-object Bernoulli process is an RFS with $b_1(\mathbf{y}), \dots, b_\nu(\mathbf{y})$ and $q_1, \dots, q_\nu$. The probability distribution takes the following form:

$$f(Y) \triangleq Q_0 \triangleq (1 - q_1) \cdots (1 - q_\nu) \quad (2.187)$$

if $Y = \emptyset$, and if $Y = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$

$$f(\{\mathbf{y}_1, \dots, \mathbf{y}_n\}) \triangleq n! \sum_{1 \leq i_1 < \dots < i_n \leq \nu} Q_{i_1, \dots, i_n} \cdot f_{i_1}(\mathbf{y}_1) \cdots f_{i_n}(\mathbf{y}_n). \quad (2.188)$$

where

$$Q_{i_1, \dots, i_n} \triangleq (1 - q_1) \cdots (1 - q_\nu) \cdot \frac{q_{i_1}}{1 - q_{i_1}} \cdots \frac{q_{i_n}}{1 - q_{i_n}}. \quad (2.189)$$

The cardinality distribution is the multi-Bernoulli distribution

$$B_{q_1, \dots, q_\nu}(n) \triangleq \sum_{1 \leq i_1 < \dots < i_n \leq \nu} Q_{i_1, \dots, i_n} \quad (2.190)$$

$$= (1 - q_1) \cdots (1 - q_\nu) \sum_{1 \leq i_1 < \dots < i_n \leq \nu} \frac{q_{i_1}}{1 - q_{i_1}} \cdots \frac{q_{i_n}}{1 - q_{i_n}} \quad (2.191)$$

$$= \left(\prod_{i=1}^n (1 - q_i) \right) \sigma_{\nu, n} \left(\frac{q_{i_1}}{1 - q_{i_1}}, \dots, \frac{q_{i_n}}{1 - q_{i_n}} \right). \quad (2.192)$$

where

$$\sigma_{\nu, n}(y_1, \dots, y_n) \triangleq \sum_{1 \leq i_1 < \dots < i_n \leq \nu} y_{i_1} \cdots y_{i_n} \quad (2.193)$$

for $0 \leq n \leq \nu$.

3. LEVEL 2 FUSION: MEASUREMENT-TO-TRACK FUSION

The level-2 fusion, or situation assessment, emphasizes more advanced inference compared to level 1. The goal is to discern probable situations based on observed events and gathered data. It connects the dots between objects, quantifying relationships like proximity and communication to gauge the importance of entities or items in a given context. The objective at this level is to draw high-level conclusions and pinpoint key activities and events, essentially recognizing patterns. The result is a collection of high-level inferences. This chapter elucidates the Multi-Target Tracking and Association (M2TA) problem, also called measurement-to-track fusion, as part of the level 1 and level 2 fusion process, and discusses how non-traditional measurements can be modeled within the Random Finite Set (RFS) framework for application in Bayesian fusion problems.

Sensors possess the ability to supply a diverse array of information and functionalities. As such, it is neither feasible nor rational to presume that a single measurement model can encapsulate these characteristics. A prevalent measurement model extensively employed in multi-target (MT) tracking applications is the point-detection model. Under this model, provided that targets are sufficiently remote from the sensor and the sensor possesses (ideally) boundless resolution, the subsequent assumptions can be made: During every scan, an individual target produces a maximum of one measurement, and each measurement is generated by only a single target [86],[89],[98]. This is denoted as the *standard measurement model*. This sensor model is capable of accommodating quantities such as position, bearing, frequency, etc., as measurements. Nevertheless, due to the restricted resolution capabilities of certain sensors, including some sonar systems, multiple targets may produce a single measurement when they occupy the same resolution cell. This is referred to as merged measurements. Furthermore, in applications like video target tracking, individual target detections comprise multiple points, termed extended targets. Such instances directly influence the tracking system's performance, necessitating a systematic investigation.

In [33], [88], Mahler explores non-traditional sensor models and posits that the statistical characterization of non-traditional or ambiguous data can be modeled employing the Random Finite Set (RFS) framework and the Finite Set Statistical

Theory (FISST) [88], [118]. Conceptually, Mahler suggests constructing generalized likelihood functions for non-traditional sensor models and data with varying degrees of ambiguity. Modeling is founded on Bayes' rule, and its extensive generality enables modeling utilizing RFSs. For instance, RFS models allow for the incorporation of pixelized image data [119] and super-positional sensors [120]—two examples of non-traditional sensors—into Bayes-optimal processing. Other instances of non-standard data encompass features, inference rules, and natural language statements [86], [121]. Employing natural language statements, such as *the target is beside the tree* or *in the vicinity of the building*, as measurements in multi-object systems is intriguing, as it facilitates the direct usage of human/operator observations. This is particularly beneficial in situations where subjective, mission-specific knowledge is required.

The chapter outline is as follows. Initially, an exhaustive review of the literature pertaining to non-traditional measurement models is provided. Subsequently, the standard point-target model, which forms the foundation of the majority of conventional tracking methods, is delineated. Following this, the general modeling methodologies of measurements, which deviate from the conventional point-target model, are introduced. These methodologies are represented in the form of generalized likelihood functions. The discussion then shifts to the cluster target tracking problem, wherein targets generate multiple measurements per scan. A mathematical model for these cluster targets is proposed. A novel Cluster-Target Probability Hypothesis Density (CT-PHD) filter is then presented, and Markov transition and measurement likelihood functions are presented in the form of probability-generating functionals. Building upon the Bayesian nonparametric modeling framework, the Dirichlet process mixture model is utilized for dynamically estimating target clusters from measurement clouds. Finally, the section culminates with the presentation of numerical results derived from real-world target trajectories. This comprehensive overview provides a detailed exploration of the various models and methodologies utilized in the field of multi-target tracking and data association.

3.1. Literature Overview

Recall that the conventional measurement model assumes that a single target can only yield one measurement. Yet, numerous practical issues don't align with this supposition. The term "cluster-type targets" is used to describe situations where a

single target might produce several measurements, a result of certain measurement generation procedures about which our knowledge may be incomplete or entirely lacking. Thus, the cluster-type targets are divided into three problems: (i) extended targets, (ii) group targets, and (iii) cluster targets. In all three cases, the number of measurements that a single target generates is (in general) arbitrary, and can change in time.

Different systems may use different sensor types which may provide various kinds of data for specific MTT applications. In this chapter, a literature review is presented for filters using different sensor types for target-tracking applications based on RFS models.

Balakumar et al. [122] considered the direction of arrival problem for sinusoidal signals to track signal sources with unknown cardinality in a non-stationary environment. They proposed a tracking algorithm for super-positional sensors based on the probability hypothesis density (PHD) filter to track both cardinality and the kinematic states of the sources jointly. Their proposed approximate likelihood provided superior performance compared to the sequential direction of arrival estimation with reversible jump Markov chain Monte Carlo particle filter. The super-positional models are required for closely spaced targets when the targets are in close proximity and the conventional detection algorithms fail to discriminate them. A general super-positional cardinalized PHD (CPHD) filter referred to as Σ -CPHD filter is derived by Mahler [106]. However, the Σ -CPHD filter is computationally intractable and needs approximate solutions. Hauschildt, as referenced in [123], put forth a Gaussian mixture approximation leading to a (partially) manageable, definitive solution for the precise Σ -CPHD. However, it is still computationally demanding and only appropriate for a small number of targets. Another approximation is proposed by Thouin et al. [124] for super-positional sensors. This filter is applied to two problems, namely, multi-target tracking using passive acoustic sensors and multi-target detection and tracking using RF tomography sensor arrays, and found to be superior to MCMC in terms of error and computational complexity.

A similar approach can be applied to pixelized image data. Conventional algorithms detect blobs in image frames and then extract features of the blobs to use as measurements. However, this often eliminates target pixels along with clutter and causes the loss of information, especially in low SNR scenarios. The hypothesis is that if the detection and tracking could have been performed jointly using entire image data, rather than the extracted features, the performance would have improved.

Vo et al. [125] proposed an image-observation multi-target multi-Bernoulli filter. Their IO measurement model draws from the RFS theory, with the filter representing a closed form and exact solution to the multi-target Bayes filter. Demonstrations have proven that it surpasses state-of-art track-before-detect filters when applied to image observation tasks.

In the extended target case, there is a single physical target and this single target is associated with more than one measurement. For example, a video target may consist of several pixels or multiple measurements can be obtained due to the scatterers on the target surface [33], [35], [44]. The parameters of extended targets may include centroid, target, and target shape (target extent). In target tracking literature, the extended targets are usually encountered in the video (or image) data and in lidar or high-resolution sensors [98]. The extended target tracking problem is inherent in areas such as autonomous vehicles, robotics, etc. and there are a plethora of studies on it. In literature, the studies on extended target tracking mostly focus on the following.

- The statistics of extended targets: it can be state-only, state and shape, and state and scatterers.
- An exact model of a rigid body depicted as a collection of point-scatterers situated at stationary points on a solid structure. [33].
- The Approximate Rigid-Body (ARB) model serves as an estimated representation of the rigid-body model [44].
- Approximate Poisson-body (APB) model was proposed by Gilholm et al. [126], which is premised on the idea that an extended target creates a set of measurements that manifest as a cluster with continuous distribution.
- In [127] Ristic et al. proposed a filter so-called Bernoulli particle filter for extended targets with unknown shapes.
- When considering the count of measurements in an extended target based on the Poisson assumption, as referenced in Gilholm's 2005 study [128], a binomial distribution is proposed, as mentioned in the 2013 work by Ristic [129].
- For modeling the object shape (or extent), [45] uses no specific shape model, Gamma Gaussian Inverse Wishart distribution for prior on the extent [130] are proposed.
- In [131] a closed-form solution for multi-static radar is proposed for extended target tracking problem, utilizing the random matrix framework.

There are some constraints on the extended targets. For instance, due to their

continuous physical forms, extended targets cannot physically intersect. However, they are capable of occluding one another [44]. Given that the measurements are solely dependent on the state of the extended target, it allows for a direct estimation of the state derived from these measurements.

Group targets are comprised of autonomous or semi-autonomous point targets that exhibit collective behavior through coordination, comparable to an infantry battalion or an aircraft sortie. The state parameters of a group target can be akin to those of an extended target, encompassing centroid, centripetal velocity, formation type, the count of targets in the group, and group-shape parameters, as suggested by [44]. Yet, unlike an extended target, determining the states of the individual components is also a requisite in the case of group targets.

Distinct from extended targets, group targets possess the ability to interpenetrate but not occlude each other, as indicated in the works [42], [132]. The measurements generated by a group target are not directly reliant on the state of the group but instead depend on the states of the individual targets within the group. Consequently, this necessitates an indirect estimation, thereby making the problem more challenging.

Within the body of target tracking literature, the issue of group target tracking has not been as extensively explored as that of extended target tracking, and the existing studies often overlap with extended target tracking [46], [48], [133]–[136].

In the third case, a single target producing multiple measurements is called a cluster target. Different than the first two cases, a cluster target is actually a point target. However, since the process of measurement generation, in this case, is completely unknown, the state of a cluster target is inferred indirectly from the measurement cluster it produces [44]. The observation process is an unknown and time-evolving random process. The problem in the cluster target case is to detect and track the targets via measurement clusters which dynamically evolve in time. Thus, it can be regarded as the generalization of the (static) clustering problem with an additional step that after clustering the measurements, they are used to update their associated tracks.

This dynamic clustering (or cluster-target tracking) problem can be handled in two ways. If the clusters are defined as disjoint sets and points are assigned to only one cluster, then this is called hard clustering. If the clusters have overlapping, fuzzy boundaries, and points can be assigned to clusters with some membership values, based on some similarity or closeness measure, then the problem is known as soft clustering. The static case of clustering is a widely studied problem and there are a plethora of works in literature, for both hard and soft cases. However,

the dynamic case is not widely studied as its static counterpart, especially in target tracking literature. Considering some sensor modalities, such as lidar, over-the-horizon radar (OTHR), passive coherent localization (PCL), or high-range resolution radar (HRRR), that possibly provide multiple measurements for a single target [85], are becoming widespread for some specific applications, such as autonomous vehicles or simultaneous localization and mapping, this dynamic clustering or cluster-target tracking problem has not drawn enough attention.

Mahler's contributions to the problem of cluster target tracking [137],[138] are notable. In [137], Mahler extended Cheeseman's Bayesian clustering algorithm [139] to accommodate dynamic cases. Further, in [138], he derived the update equations for both the PHD and CPHD filters with respect to the multi-object Poisson likelihood function.

This proposed methodology embodies a soft clustering approach, capable of estimating the quantity of clusters, their shapes, and relative densities. When formulating the update equations for the PHD and CPHD filters, Mahler hypothesized the existence of a Markov transition density for the cluster target parameters.

Nonetheless, no such density has been proposed in the existing literature. Therefore, a Markov density that accurately models the time evolution of individual parameters must be initially defined [138]. Furthermore, despite Mahler's conceptual proposal, no numerical implementation specific to the problem at hand was suggested.

3.2. Standard Measurement Model: Point-Target

Most of the estimation algorithms in target tracking literature consider the so-called point-target model, in which it is assumed that the targets are distant enough to detect as a single-point target. In other words, every target generates only one observation [86]. In addition, the model assumes that any observation can only be generated by a single target. This makes it possible to classify a measurement either as originated by a real target or as a false alarm (or clutter, noise). If an existing target has not been detected, it is referred to as a missed detection.

Through this presumption, the likelihood function can be developed using the provided measurement model. Let us be given a measurement of the form $z = h(x) + w$, where w is a zero-mean noise with probability density $f_W(w)$. The measurement likelihood function is denoted as $f(z|x) = f_{Z|X}(z|x)$, which describes the probability

of measurement z , given the true state x . Starting with the probability mass function $p(S|x) \triangleq p(z|x) = Pr(Z \in S)$, the total probability is

$$p(E_z|x) = \int_{E_z} f(y|x) dy \cong f(z|x) \lambda(E_z) \quad (3.1)$$

where E_z is a small region around the point z with hyper-volume $\lambda(E_z)$. Then the likelihood function, in the limit case, is

$$f(z|x) = \frac{\partial p}{\partial z} \triangleq \lim_{\lambda(E_z) \rightarrow 0} \frac{p(E_z|x)}{\lambda(E_z)} \quad (3.2)$$

Then, by using the additive noise model, the likelihood function can be written as

$$f(z|x) = \lim_{\varepsilon \rightarrow 0} \frac{Pr(Z \in B_{\varepsilon,z})}{\lambda(B_{\varepsilon,z})} = \lim_{\varepsilon \rightarrow 0} \frac{p_W(B_{\varepsilon,z-h(x)})}{\lambda(B_{\varepsilon,z-h(x)})} = f_W(z - h(x)). \quad (3.3)$$

The resulting formula in (3.3) can serve as the likelihood function of detected targets within the Bayesian filtering framework.¹

3.3. Generalized Data Modeling

Many forms of (conventional) data can be described in statistical form. However, there are many more statistically uncharacterizable data forms that are encountered in real-world problems, such as natural language statements, features, rules, etc. The information having ambiguous forms includes additional uncertainties other than randomness. These uncertainties make it harder to model such data. This is one of the most challenging aspects of data fusion. The problem of data having different ambiguities (or uncertainties) must be handled in formal Bayes modeling, so that unconventional information types can be used in an information fusion framework, in a Bayes-optimal manner.

In this section, the methodology for extending the formal (conventional) Bayes modeling to the case of the unconventional measurements is presented as mentioned in [86]. This modeling approach, in fact, will be the backbone of incorporating non-traditional measurements into the target tracking. As Mahler stated in [118],

¹The expression given in (3.3) can be used in the single-target tracking, with perfect detection model, and without clutter. For MTT with clutter and missed detections, more sophisticated models should be used.

[140], such a scheme is a potential (conceptual) bridge between probabilistic and non-probabilistic forms of expert-systems reasoning, and it can bring hard and soft information fusion to a common ground.

In the sense of information fusion, there are two types of data, namely, state estimate and measurement; and four types of measurements, as Mahler, discriminated in [86]:

- Measurements that are unambiguously generated and are unambiguous (UGU),
- Measurements that are unambiguously generated but are ambiguous (UGA),
- Measurements that are ambiguously generated and are ambiguous (AGA),
- Measurements that are ambiguously generated but are unambiguous (AGU).

UGU measurements are conventional measurements, which are modeled by the techniques discussed in the previous section. AGU measurements are the measurements that have statistically uncharacterizable variations. The likelihood functions of these measurements are ambiguous, though the measurements themselves are not. For UGA measurements, the mathematical relation between the measurements and the state is precisely known. In other words, the likelihood function of UGA measurements contains uncertainties stemming only from randomness. But the observed object (or target) is ambiguous, i.e. rules and natural language statements, or features extracted (by human operators) from signatures. In AGA measurements, both the observed phenomenon and its relationship to the target state (likelihood function) are ambiguous.

The formal mathematical data modeling can be summarized as follows:

- Determine the data type of the data D that is being considered: measurement or state estimate.
- If a measurement, then what type: UGU, UGA, AGA, AGU.

After determining data type,

- Specify the uncertainty modeling method suitable for the problem: Fuzzy logic, Dempster-Shafer, etc.
- Convert this into a random set representation D .
- Employ this depiction to formulate a generalized likelihood function $f_{k+1}(D|x) = f_{k+1}(D|x)$.
- Apply the data fusion procedure by utilizing joint generalized likelihood functions during the correction phase of the Bayes filter.

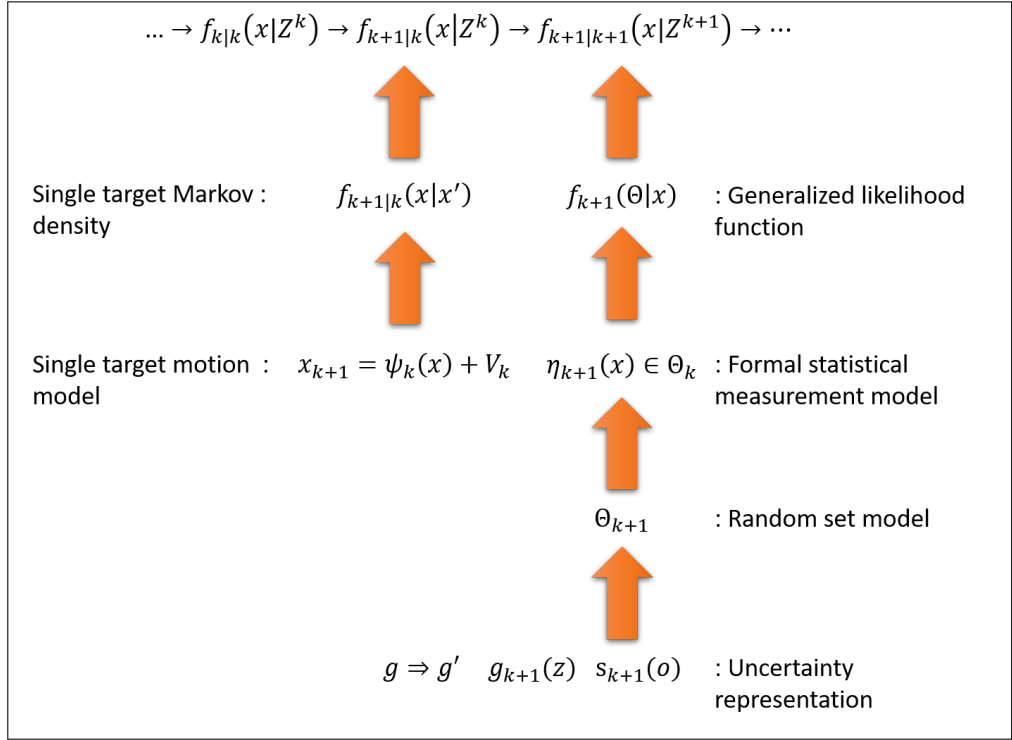


Figure 3.1: Bayesian processing of generalized measurement data.

Fig. 3.1 shows the formal (Bayesian) processing of the generalized data. As can be seen from the figure, the modeling of the data (measurements) is important and depends on the type of data used. The data is converted to a random set representation and using this representation, a generalized likelihood, which conveys statistical information in the data is constructed. This likelihood is subsequently employed in the correction phase of the Bayes filter to derive the posterior distribution of the target state considering the historical measurement data.

3.3.1. Generalized Likelihood Functions

In [86], Mahler extends the traditional Bayesian modeling applied to standard measurements to include unconventional measurements and various other data types. Examples of such measurements are [33]

- Quantized measurements are typically used for data compression.
- Attributes, such as those extracted by a human operator from an image captured by a camera.
- Features, which are derived from a sensor signature using a specific algorithm.
- Statements in natural language, which could be provided by a human or extracted

from a textual document.

- Inference rules that are obtained from a knowledge database.

By following the methodology of generalized data modeling and FISST calculus, different kinds of observations from different sources can be incorporated into the tracking and data fusion process. For example, since the sensors usually provide quantized measurements (because of limited resolution or communication requirements), the actual measurement has another ambiguity besides noise. The quantized measurement is imprecise, besides being uncertain. Different ambiguities in measurements can be generalized as *generalized measurements*, and the statistical characteristics of these generalized measurements can be described by *generalized likelihood functions*. Usually, this uncertainty is modeled by a nonlinear-additive measurement model [33]:

$$Z = \eta(x) + V \quad (3.4)$$

Then, the corresponding likelihood function is

$$f(z | x) = f_V(z - \eta(x)) \quad (3.5)$$

However, z is only a mathematical model z_ζ of some real-world measurement ζ . Thus, the actual likelihood is of the form

$$f(\zeta | x) = f(z_\zeta | x). \quad (3.6)$$

In [33] Mahler presents an example of non-traditional measurement for quantized measurement. Measurements are usually quantized for bandwidth limitations or due to the limited sensor resolution. Suppose that z is a measurement vector in a measurement space Ξ . Let Ξ be partitioned into cells, i.e. quantized, that are mutually disjoint $T_1 \uplus \dots \uplus T_m = \Xi$. The quantized measurement is the entire subset T_i . Accepting T_i as a measurement is equivalent to saying that we are unable to specify the actual measurement, but the interval that it is contained $z \in T_i$. In other words, the quantized measurement is imprecise.

Given a nonlinear additive measurement model, $Pr(Z \in T_i)$ can be explicitly computed:

$$\begin{aligned}
Pr(\eta(x) + V \in T_i) &= \int_{T_i - (x)} f_V(z) dz \\
&= \int_{T_i} f_V(z - \eta(x)) dz \\
&= \int_{T_i} f(z|x) dz
\end{aligned} \tag{3.7}$$

where $T_i - (x) \triangleq \{z - \eta(x) \mid z \in T_i\}$. Suppose that Θ of Ξ is a generalized measurement and let $Z = \eta(x) + V$ a nonlinear additive measurement model. Then, $Z \in \Theta$ is a generalized measurement model. The function

$$\rho(\Theta|x) \triangleq Pr(Z \in \Theta|x) = Pr(\eta(x) + V \in \Theta). \tag{3.8}$$

is called a *generalized likelihood function (GLF)* [33]. The GLF provides the capacity to transfigure one uncertainty representation into another, in a manner that is empirically Bayes-optimal. This results in a rigorous strategy to account for the unreliability of information sources.

3.4. Cluster-Target Tracking

Data clustering, which is a widely used method in the areas such as pattern recognition and machine learning, is the problem of partitioning a dataset into an unknown number of subsets that are mutually disjoint, based on a measure of similarity or proximity between the points. In the mentioned areas, the problem is almost always static, in other words, the statistics of the clusters, such as centroid, shape, cardinality, etc. do not change in time. In target tracking, however, due to the dynamic nature of the target and data generation processes, the time evolution of the measurement clusters must be considered.

Due to some unknown, time-evolving random process, a single sensor may be collecting multiple measurements from individual targets. Suppose that the measurement vectors collected at each scan are denoted as $z \in \mathfrak{Z}_0$, where $\mathfrak{Z}_0$ is the measurement space. The sensor provides a time series of measurements denoted as $Z^{(k)} : Z_1, \dots, Z_k$ where k is the time index and $Z_j = \{z_1, \dots, z_m\}$ is the measurement set collected at time j . However, the measurement generation process is unknowable, hence we assume that it can be (approximately) modeled as a Poisson mixture, i.e.

$$f_{k+1}(Z|X) = e^{-(a_1+\dots+a_n)} \prod_{z \in Z} \theta_{k+1}(z|X) \quad (3.9)$$

where

$$X = \{(a_1, x_1), \dots, (a_n, x_n)\}. \quad (3.10)$$

and where $a_1, \dots, a_n \geq 0$ and where X is called the model, x_i is the parameter vector of an individual target, and n is the number of targets (or clusters). And the intensity function (PHD) of the Poisson process $\theta_{k+1}(z|X)$ is defined as

$$\theta_{k+1}(z|X) = \frac{a_1 \theta_{k+1}(z|x_1) + \dots + a_n \theta_{k+1}(z|x_n)}{a_1 + \dots + a_n} \quad (3.11)$$

where $\theta_{k+1}(z|x)$ is the likelihood of an individual target and where the mixture weights are defined as

$$\omega_i = \frac{a_i}{\sum_i a_i}. \quad (3.12)$$

which indicate the contribution of each individual *model* to the generation of Z . Thus, the aim is to estimate the parameters $n, x_1, \dots, x_n$, and $a_1, \dots, a_n$. This can be achieved in a Bayes-optimal manner via the (multi-object) Bayes filter recursion:

$$\dots \rightarrow f_{k|k}(X|Z^{(k)}) \rightarrow f_{k+1|k}(X|Z^{(k)}) \rightarrow f_{k+1|k+1}(X|Z^{(k+1)}) \rightarrow \dots$$

This requires that there exists a Markov transition density for the cluster-target parameters in the following form:

$$f_{k+1|k}(a, x|a', x') \quad (3.13)$$

However, to the best of our knowledge, there is no such practical transition density model in the literature. Thus, we should design a Markov transition density that suits the problem specifications at hand. In what follows, we will present the derivation for this density to model the time evolution of the individual models.

3.4.1. Markov Transition Density for Cluster-Target Model

The multi-cluster model includes various parameters. We want to estimate the model parameters, namely, cluster size, cluster kinematics, and cluster shape. In

previous section, the model was given as $X = \{(a_1, x_1), \dots, (a_n, x_n)\}$, where a_i is the cluster size and $x_i = (r_i, R_i)$ consists of cluster centroid r and shape variables positive semi-definite matrix R , respectively.

$$a = \sum_i a_i$$

gives the expected number of observations in the model.

The time-update formula, i.e. the Chapman-Kolmogorov equation for the multi-target motion is given in (2.59). If we rewrite (2.59) for the cluster-target model given in (3.10), we obtain

$$f_{k+1|k}(a, x|z^{(k)}) = b_{k+1}(a, x) + p_s \int f_{k+1|k}(a, x|a', x') f_{k|k}(a', x'|Z^{(k)}) da' dx'. \quad (3.14)$$

where $b_{k+1}(a, x)$ is the spontaneous birth intensity and $f_{k|k}(a', x'|Z^{(k)})$ is the posterior at time k .

When the multi-target density is assumed to be a Poisson point process, then the predicted multi-target density can be described by the PHD². Thus, (3.33) can be written in terms of PHD as follows:

$$D_{k+1|k}(\xi|Z^{(k)}) = D_{k+1}^b(\xi) + \int p_s f_{k+1|k}(\xi_{k+1}|\xi_k) D_{k|k}(\xi|Z^{(k)}) d\xi_k. \quad (3.15)$$

where $\xi = (a, r, R)$ is the composite cluster state.

As can be seen from (3.15), we want to process non-kinematic information along with kinematic information. Here, we make an assumption to arrive at a closed-form expression for the solution of the Chapman-Kolmogorov integral in (3.14). We assume that kinematic and non-kinematic information is (approximately) statistically independent. Then, we can separate the kinematic and non-kinematic parts and rewrite (3.14) as

$$D_{k+1|k}(a, x|Z^{(k)}) \cong D_{k+1}^b(a, x) + p_s \left(\int f_{k+1|k}(a|a') D_{k|k}(a'|Z^{(k)}) da' \right) \times \left(\int f_{k+1|k}(x|x') D_{k|k}(x'|Z^{(k)}) dx' \right). \quad (3.16)$$

The first integral in (3.16) describes the time evolution of the cluster-size parameter. The second integral describes the time evolution of the cluster centroid, as well as

²See (2.82-2.94)

the cluster shape. Thus, we approximate the time update of the cluster-target model as the product of the time update of the kinematic part and the time update of the non-kinematic part. Now, we can rewrite (3.14) as

$$D_{k+1|k}(a, x|Z^{(k)}) = D_{k+1}^b(a, x) + p_s D_{k+1}(a|Z^{(k)}) D_{k+1}(x|Z^{(k)}). \quad (3.17)$$

From now on, we can design a Markov transition density for each part separately to arrive at a closed-form expression for the time-updated density of the cluster-target model. We begin by defining the Markov transition density for the kinematic part.

We defined the cluster-target state as $x = (r, R)$ where r is the kinematic state of the cluster centroid and R is the shape parameter as a positive semi-definite matrix. Borrowing from the extended target tracking concept, we model the shape parameter as a random matrix, in which the shape of the object is defined as a symmetric positive-semi definite matrix and is estimated as a part of the object state. The Chapman-Kolmogorov integral for the kinematic part is given as follows:

$$D_{k+1|k}(r, R|Z^{(k)}) = \int f_{k+1|k}(r, R|r', R') D_{k|k}(r', R'|Z^{(k)}) dr' dR'. \quad (3.18)$$

In (3.18), $D_{k|k}(r', R'|Z^{(k)})$ is the posterior PHD for the kinematic part at time step k . First, we write the transition density and the posterior density of the cluster center r conditioned on the previous state, respectively:

$$f_{k+1|k}(r, R|r', R') = f_{k+1|k}(r|R, r', R') f_{k+1|k}(R|r', R'), \quad (3.19)$$

$$D_{k|k}(r', R'|Z^{(k)}) = D_{k|k}(r'|R', Z^{(k)}) D_{k|k}(R'|Z^{(k)}) \quad (3.20)$$

Then, we make a simplifying assumption, along with usual Markov assumptions, that the shape of the cluster target is statistically independent of the target kinematics, but the evolution of the kinematics depends (only) on the (predicted) cluster shape, given the previous target state. Then, the Markov density in (3.19) can be rewritten as follows:

$$f_{k+1|k}(r, R|r', R') = f_{k+1|k}(r|R, r') f_{k+1|k}(R|R'). \quad (3.21)$$

Applying these assumptions to (3.18) we get the (approximate) time-updated density for the cluster parameters:

$$\begin{aligned}
D_{k+1|k}(r, R|Z^{(k)}) &\cong \int f_{k+1|k}(r|R, r') f_{k+1|k}(R|R') \\
&\quad \times D_{k|k}(r'|R', Z^{(k)}) D_{k|k}(R'|Z^{(k)}) dr' dR'
\end{aligned} \tag{3.22}$$

By assuming that the time evolution of the cluster shape does not affect the evolution of the kinematics, the time-updated density for the cluster parameters can be written as the product of two independent integrals.

$$\begin{aligned}
D_{k+1|k}(r, R|Z^{(k)}) &= \int f_{k+1|k}(r|R, r') D_{k|k}(r'|R', Z^{(k)}) dr' \\
&\quad \times \int f_{k+1|k}(R|R') D_{k|k}(R'|Z^{(k)}) dR'.
\end{aligned} \tag{3.23}$$

Then, we can obtain an expression for the time-updated (predicted) density of the kinematics and shape as

$$D_{k+1|k}(r, R|Z^{(k)}) = D_{k+1|k}(r|R, Z^{(k)}) D_{k+1|k}(R|Z^{(k)}). \tag{3.24}$$

(3.24) indicates that the time-updated density for the joint kinematics and cluster shape can be obtained by marginalizing over the time-updated cluster shape parameter. This allows us to model the time evolution of the shape parameter independent of the kinematics and greatly simplifies the calculations.

When the distributions for the kinematic and extension state are assumed to be Gaussian and inverse Wishart distributed, respectively, they become conjugate prior to the cluster target model. Then the cluster state distribution can be written as follows:

$$D_{k|k}(r, R|Z^{(k)}) = D(r_k|R_k, Z^{(k)}) D(R_k|Z^{(k)}) \tag{3.25}$$

$$= \mathcal{N}(r_k; m_{k|k}, P_{k|k} \otimes R_k) \times \mathcal{IW}(R_k; \nu_{k|k}, V_{k|k}). \tag{3.26}$$

This hybrid distribution is called *Gaussian inverse Wishart* (GIW) density, where the $\nu_{k|k}$ is the degrees of freedom and $V_{k|k}$ is the scale matrix. Due to this conjugacy, the time prediction of kinematic and extension states preserves the *GIW* distribution, thus providing a tractable and mathematically rigorous time update formulae for the cluster-object transition. The model can be further simplified by assuming that the measurement spread (i.e. the object extent) is dominant with respect to the measurement noise.

The time update for the non-kinematic part is simpler. The non-kinematic part consists of the cluster size, i.e. number of points in the cluster and it is a quantity that is related to the measurement generation process. Since we only propagate a target track from time k to $k+1$, we assume the cluster size has not changed. Thus, we can express the Markov transition density of the cluster size as

$$f_{k+1|k}(a|a') = \delta_a(a'). \quad (3.27)$$

where $\delta_a(a')$ is the Dirac delta distribution concentrated on a' . Then, the time-updated density for the non-kinematic part can be written as

$$f_{k+1|k}(a|Z^{(k)}) = \int f_{k+1|k}(a|a') f_{k|k}(a'|Z^{(k)}) da' \quad (3.28)$$

$$= \int \delta_a(a') f_{k|k}(a'|Z^{(k)}) da' \quad (3.29)$$

$$= f_{k|k}(a|Z^{(k)}). \quad (3.30)$$

Recall that, the cluster measurement process is modeled as a Poisson point process (3.9). Thus, the model depends on the Poisson rate a_i , which is generally unknown. To simplify the modeling process, we model the unknown rate parameter as a beta-distributed random variable, which is the Bayesian conjugate prior to the Poisson rate. Then the predicted density in (3.30) is approximately modeled as follows:

$$D_{k+1|k}(a|Z^{(k)}) = \mathcal{G}(a; \alpha_{k+1|k}, \beta_{k+1|k}). \quad (3.31)$$

Finally, by substituting (3.24) and (3.30) into (3.17), we can write the time-updated density for the cluster parameters as

$$D_{k+1|k}(a, r, R|Z^{(k)}) = D_{k+1|k}(a|Z^{(k)}) D_{k+1|k}(r|R, Z^{(k)}) D_{k+1|k}(R|Z^{(k)}) \quad (3.32)$$

The resulting equation in (3.32) is based on a product form of individual model parameters, given the independence assumption between them. In other words, the predicted density can be written in a factored form. Thus, the time prediction of the model variables can be modeled individually, simplifying the process. The predicted cluster-target density in (3.32), given the previous modeling assumptions in (3.26)

and (3.31), is called *Gamma-Gaussian inverse Wishart* density and can be written as follows:

$$D_{k+1|k}(a, r, R|Z^{(k)}) = \mathcal{G}(a; \alpha_{k+1|k}, \beta_{k+1|k}) \mathcal{N}(r_k; m_{k|k}, P_{k|k} \otimes R_k) \mathcal{IW}(R_k; \nu_{k|k}, V_{k|k}). \quad (3.33)$$

In the next section, we will present the Bayes update formulas for the cluster-target model defined by (3.9).

3.4.2. Bayes Update for the Cluster-Target Model

The multi-target analog of the recursive Bayes filter, as expressed in (2.1) and (2.2), typically presents computational intractability. Consequently, numerical approximations are essential for the practical implementation of these filters in real-world scenarios. The cluster-target problem is much more demanding in that regard. The cluster-target tracking is a combinatorial problem in its nature and it is also computationally intractable. Tractable approaches for algorithmically implementing the cluster-target tracking filter are a subject of research. Thus, in this section, we present PHD filter update equations for the cluster-target model given above. These equations were derived by Mahler, however, no numerical algorithm or simulations were proposed in that work. We propose Gaussian mixture and sequential Monte Carlo implementations for the cluster-target problem and evaluate them in simulated scenarios.

In the previous section, the time-update formulas were presented for the cluster-target problem and a Markov transition density describing the time evolution of the parameters was given for the problem.

We begin by giving a brief review of the PHD filter for the standard measurement and motion models, then we extend the equations for the cluster-target model defined in the previous section. The probability hypothesis density (PHD) $D_{k|k}(\xi) = D_{k|k}(\xi|Z^{(k)})$ is a density function that is defined on single-target state vectors and is an approximation of the multi-target density. The PHD is mathematically defined as an expected value of the multi-target density function, given by the following formula:

$$D_{\Xi} \triangleq \mathbb{E}[\delta_{\Xi}(\xi)] = \int \delta_X(\xi) f_{\Xi}(X) \delta X. \quad (3.34)$$

where $\delta_w(\xi)$ is the Dirac delta density concentrated on w , Ξ is the random finite set. The Probability Hypothesis Density (PHD) constitutes the first moment of the multi-target density, providing an estimation of the number of targets within a given region. The PHD filter carries forward this density over time. In particular,

$$N_{k|k} \triangleq \int D_{k|k}(\xi) d\xi. \quad (3.35)$$

The PHD filter predictor equation for the standard motion model is given by the equation

$$D_{k+1|k}(\xi|Z^{(k)}) = b_{k+1|k}(\xi) + p_s \int f_{k+1|k}(\xi|\xi') D_{k|k}(\xi|Z^{(k)}). \quad (3.36)$$

In PHD filter, it is assumed that the predicted multi-target distribution $f_{k+1|k}(X|Z^{(k)})$ is at least approximately a Poisson process [86]. Then, the PHD updated with the measurement set at time k , Z_{k+1} , is given by the approximate equation

$$D_{k+1|k+1}(\xi|Z^{(k+1)}) \cong L_{Z_{k+1}}(\xi|Z^{(k)}) D_{k+1|k}(\xi|Z^{(k)}). \quad (3.37)$$

where the likelihood function $L_Z(\xi|Z^{(k)})$ is

$$L_{Z_{k+1}}(\xi|Z^{(k)}) = 1 - p_D + \sum_{z \in Z} \frac{p_D L_z(\xi)}{\lambda_{k+1} c_{k+1}(z) + D_{k+1|k}[p_D L_z]}. \quad (3.38)$$

Alternatively, the prediction and update equations of the PHD filter can be written in the form of probability-generating functional (p.g.fl.). The p.g.fl. of a multi-target density is defined as

$$G_f[h] = \int h^Y f(Y) \delta Y \quad (3.39)$$

where $h^Y = 1$ if $Y = \emptyset$ and $h^Y = \prod_{y \in Y} h(y)$ if otherwise and $h(y)$ is a non-negative test function. Then, the multi-target time update equation in (2.1) can be written in p.g.fl. form as

$$G_{k+1|k}[h] = \int G_{k+1|k}[h|\xi'] f_{k|k}(\xi'|Z^{(k)}) \delta \xi'. \quad (3.40)$$

where

$$G_{k+1|k}[h|\xi'] \triangleq \int h^\xi f_{k+1|k}(\xi|\xi') \delta\xi. \quad (3.41)$$

is the p.g.fl. of the multi-target Markov transition density $f_{k+1|k}(\xi|\xi')$. The p.g.fl. of the standard motion model in (3.41) is

$$G_{k+1|k}[h|\xi'] = (1 - p_S + p_S p_h)^{\xi'} \cdot e^{\mu b[h] - \mu}. \quad (3.42)$$

where $b[h] \triangleq \int h(\xi) b(\xi) d\xi$ and $p_h(\xi') \triangleq \int h(\xi) f_{k+1|k}(\xi|\xi') dx$. Using (3.42) and (3.40) we get the p.g.fl. $G_{k+1|k}[h]$ of the time updated multi-target distribution as

$$G_{k+1}[h] = e^{\mu b[h] - \mu} G_{k|k}[1 - p_S + p_S p_h]. \quad (3.43)$$

In a similar way, we can write the Bayes update formula in (2.2) for the multi-target distribution in p.g.fl. form. The p.g.fl. form for (2.2) is

$$G_{k+1|k+1}[h] = \frac{\frac{\delta F}{\delta Z_{k+1}}[0, h]}{\frac{\delta F}{\delta Z_{k+1}}[0, 1]} \quad (3.44)$$

where

$$F[g, h] \triangleq \int h^X G_{k+1|k}[g|X] f_{k+1|k}(X) \delta X \quad (3.45)$$

and

$$G_{k+1|k}[g|X] \triangleq \int g^Z f_{k+1|k}(Z|X) \delta Z \quad (3.46)$$

is the p.g.fl. of the (standard) multi-target likelihood $f_{k+1}(Z|X)$. The closed form expression for the $G_{k+1}[g|X]$ of the standard multi-target measurement model is

$$G_{k+1}[g|X] = (1 - p_D + p_D p_g)^X e^{\lambda c[g] - \lambda} \quad (3.47)$$

where $c[g] \triangleq \int g(z) c(z) dz$, and $p_g(x) \triangleq \int g(z) f_{k+1}(z|x) dz$, the p.g.fl.s for the clutter distribution and the likelihood, respectively. Then, using these expressions, we can write

$$F[g, h] = e^{\lambda c[g] - \lambda} G_{k+1|k}[h(1 - p_D + p_D p_g)]. \quad (3.48)$$

Again, the measurement-updated multi-target distribution at time-step $k + 1$ $f_{k+1|k+1}(X|Z^{(k+1)})$ can be obtained by applying the functional derivative to the result of (3.44).

This form of multi-target prediction and correction equations allows us to derive concrete formulas for real-world problems and simplifies the predictor and corrector calculations.

The prediction and update equations for the standard motion and measurement models can be expanded to encompass the case of cluster targets. In the following part of this section, we will showcase these equations in the context of the cluster-target model delineated in the preceding section.

Now, for the defined cluster-target model in (3.9), we can compute the p.g.fl. of the cluster-target likelihood function $\dot{G}_{k+1}[g|X]$ as follows:

$$\dot{G}_{k+1}[g|X] = \int g^Z f_{k+1}(Z|X) \delta Z \quad (3.49)$$

$$= e^{\eta_g(a_1, x_1)} \dots e^{\eta_g(a_n, x_n)} = (e^{\eta_g})^X \quad (3.50)$$

where $\eta_g(a, x) = -a + a \int g(w) \theta(w|x) dw$. Then, (3.47) becomes

$$F[g, h] = \int h^X \dot{G}_{k+1}[g|X] f_{k+1|k}(X|Z^{(k)}) \delta X \quad (3.51)$$

$$= \int h^X (e^{\eta_g})^X f_{k+1|k}(X|Z^{(k)}) \delta X \quad (3.52)$$

$$= \dot{G}_{k+1}[h e^{\eta_g}]. \quad (3.53)$$

Now, assuming the predicted p.g.f. is Poisson $G_{k+1|k}(\xi) e^{N_{k+1|k}(\xi-1)}$, results in the measurement update equation for the cluster-target PHD (CT-PHD) filter is given by [33]

$$\frac{D_{k+1|k+1}(a, \xi|Z^{(k+1)})}{D_{k+1|k}(a, \xi|Z^{(k)})} = e^{-a} \sum_{\mathcal{P} \boxplus Z_{k+1}} \omega_{\mathcal{P}} \left(1 + \sum_{W \in \mathcal{P}} \frac{a^{|W|} \prod_{z \in W} \theta_{k+1}(z|\xi)}{\tilde{\tau}_W} \right) \quad (3.54)$$

where

$$\tilde{\tau}_W = \int \int_0^\infty e^{-a} a^{|W|} \left(\prod_{z \in W} \theta_{k+1}(z|\xi) \right) D_{k+1|k}(a, \xi) da d\xi, \quad (3.55)$$

$$\omega_{\mathcal{P}} = \frac{\prod_{W \in \mathcal{P}} \tilde{\tau}_W}{\sum_{\mathcal{P}' \boxplus Z_{k+1}} \prod_{W' \in \mathcal{P}'} \tilde{\tau}_{W'}}. \quad (3.56)$$

3.5. Bayesian Nonparametric Modeling: Dirichlet Processes

In (3.54) $\mathcal{P}$ is a partition that is a subset of all (non-empty) subsets of the measurement set Z_{k+1} . The union of all the elements W of $\mathcal{P}$ results in Z_{k+1} . The summation in (3.54) is over all the possible partitions of Z_{k+1} , thus, the problem is combinatorial and it is not feasible to enumerate all possible partitioning, especially when the measurement set is large. Thus, the summation over the partitions in (3.54) should contain a small subset of the possible partitions.

This is an important problem since the accuracy and effectiveness of the algorithm would depend on the accuracy of the measurement partitioning. Therefore, there is a need for a computationally tractable partitioning method without knowing the number of partitions beforehand. The Bayesian non-parametric framework seems to provide such a method. Dirichlet processes [141]–[143] may provide an alternative way to partition the data without the knowledge of the number of partitions a priori.

Dirichlet processes (DPs) are a class of stochastic processes used in Bayesian nonparametric modeling. They are often used to model complex, infinite-dimensional distributions, such as mixture models or clustering problems.

At a high level, a DP is a distribution over distributions. Specifically, a DP is a distribution over probability measures, where each probability measure is itself a distribution over some space. The DP is parameterized by a base distribution, often denoted as H , and a concentration parameter, denoted as α . The base distribution H is typically a distribution from a known family of distributions, such as a Gaussian distribution. The concentration parameter α determines the degree of clustering in the resulting distribution.

The key property of a DP is that it has a rich-get-richer property, which means that as more samples are drawn from a DP, the probability mass becomes increasingly concentrated around a smaller set of points, leading to clustering behavior. Specifically, given a base distribution H and a concentration parameter α , a sample from a DP is a discrete distribution over the support set of H . The probability mass of this distribution is distributed according to a stick-breaking process, where each "stick" represents the probability of selecting a new cluster.

One way to generate a sample from a DP is to use the Chinese Restaurant Process (CRP)[50], which is a popular construct used in Bayesian nonparametric modeling. In the Chinese Restaurant Process (CRP), envision a scenario where a customer enters a restaurant equipped with an infinite quantity of tables. The first customer chooses to

sit at the initial table. The n th customer then decides to sit either at an already occupied table, with a probability proportional to the existing customer count at that table, or at a fresh table, with a probability proportional to the concentration parameter α . The ensuing seating arrangement represents a sample from a Dirichlet Process (DP).

DPs have many applications in machine learning, such as clustering, topic modeling, and natural language processing. They provide a flexible way to model complex distributions without assuming a fixed number of components and allow for clustering behavior that can naturally capture the variability and complexity of real-world data.

In principle, via DP we estimate a *multi-target measurement distribution* over the space of cardinality and spatial domain. The DP estimates the clusters in an iterative manner. Thus, it needs a stopping criterion to terminate the process. We employ Hölder divergence (see Ch. 4 for more details) which is an information divergence method, to measure the information difference between subsequent multi-target measurement distributions (i.e. estimated clusters). When the information gain becomes insignificant, the process stops, and obtained clusters are used as the measurement partitions to be used in measurement fusion.

3.6. Numerical Studies

We present the results of the proposed cluster fusion algorithm on two different datasets. One is a synthetically generated scenario and the second is obtained through real flight data. The synthetically generated scenarios contain two different configurations: one is a four-target scenario and the other includes crossing targets, moving in a straight line and the targets enter/leave the scene at different time instants. There are clutter measurements that are Poisson distributed with rate $\lambda = 10$ and uniformly distributed within the surveillance area. Moreover, the detection probability is less than unity which means that there are missed detections. Both synthetic scenarios are run for 100 Monte Carlo runs, and the results are presented as the mean of the runs.

3.6.1. Synthetic Scenario-1

In this scenario, the measurement generation process is time-invariant. This means that, at each sampling time, the targets generate identical measurement clusters, i.e. only the cluster center is shifted, while the cluster spread and the measurement rate are fixed. In Fig. (3.2) the four target trajectories are shown in a 2-D surveillance

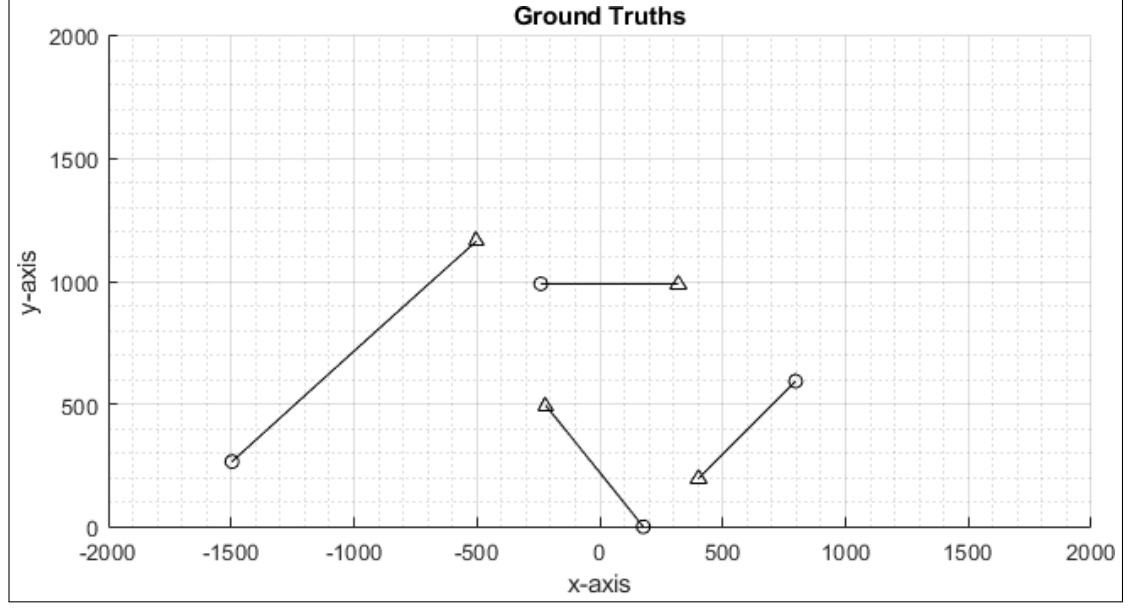


Figure 3.2: Ground-truth (synthetic) target trajectories.

area. The targets move in different directions in a linear, constant velocity motion model. They enter and leave the scene at different time instants, so the number of targets dynamically changes during the simulation.

The target state is defined in the Cartesian coordinates and includes the kinematic variables:

$$\xi_k = \begin{bmatrix} p_x & p_y & \nu_x & \nu_y \end{bmatrix}^T \quad (3.57)$$

where the state vector components are position and velocity in each axis. The measurement time interval is $T = 1s$ and the scenario duration is $100s$. According to the constant velocity motion model, the mathematical model is

$$x_{k+1} = Fx_k + w_k. \quad (3.58)$$

where w_k is the zero-mean Gaussian process noise with covariance matrix Q_k and F is the transition matrix defined as follows:

$$F = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \otimes \mathbf{I}_2 \quad (3.59)$$

$$Q = q \begin{bmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{bmatrix} \otimes \mathbf{I}_2 \quad (3.60)$$

where $\otimes$ is the Kronecker product, q is the power spectral density of the process noise.

The measurement model is also linear with the following mathematical definition:

$$z_{k+1} = Hx_{k+1} + \eta_k \quad (3.61)$$

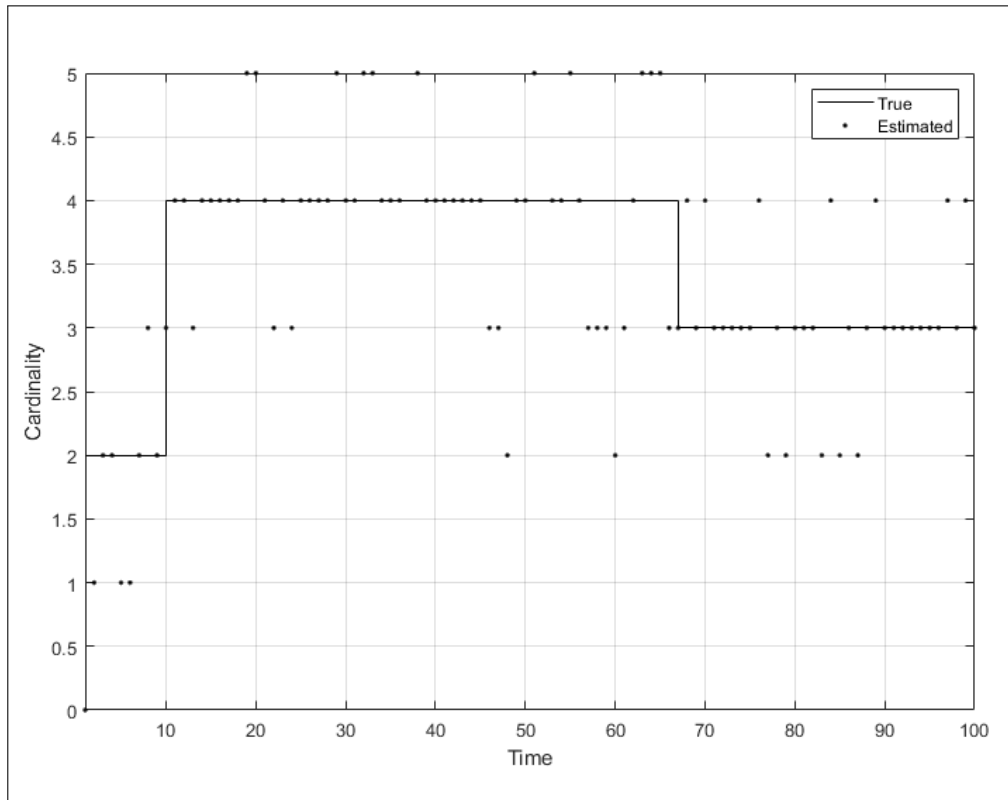
where $z = [p_x, p_y]^T$ is the measurement vector containing position information, η_k is the zero-mean Gaussian measurement noise with covariance matrix R , and H is the measurement matrix defined as follows:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (3.62)$$

$$\eta_k \cong \mathcal{N}(\eta_k; 0, R) \quad (3.63)$$

The implemented CT-PHD filter is applied to the described scenario and the results are presented in the following figures. The scenario starts with two targets in the surveillance area. At discrete time instant $k = 10$, the other two targets enter the scene. The four targets remain in the scene for most of the scenario duration. In Fig. (3.3) the cardinality estimates of the CT-PHD filter are shown. As can be seen from Fig. (3.3), the cardinality estimation is consistent, while there are occasional false alarms and missed tracks. However, these are mostly isolated events, and the targets have been tracked successfully. It should also be noted that the PHD filter has a well-known cardinality estimation problem, meaning that it has a high cardinality variance in general. Therefore, it can be said that the CT-PHD filter produced accurate results.

In Fig. (3.4), the measurements and estimated tracks are plotted against time, for each axis. Fig. (3.4) also shows the ground truth tracks in, in each axes, as a line. The gray crosses indicate the collected measurements (mean of clusters) per axis, as well



as clutter points. As can be seen from the figure, the existence of clutter measurements makes the cluster tracking problem particularly challenging.

In Fig. (3.5), the estimated track points are plotted against the ground truth trajectories in the surveillance area. The extracted point estimates are shown in red. Fig. (3.6) shows the performance of the CT-PHD filter in terms of OSPA metric. It can be inferred that the localization error is around $40m$ on average and the cardinality error is around $20m$.

3.6.2. Synthetic Scenario-2

In this scenario, the targets move in closer proximity to each other and in different directions. And there also occurs a target crossing at some time. Fig. (3.7) shows the ground truth trajectories of the targets. The targets start their movement at the positions indicated by circles ($\circ$) and end their movement at the positions indicated by triangles ($\triangleleft$) in the figure.

In Fig. (3.8) the result of the application of the proposed CT-PHD filter is shown. The true target trajectories are shown in black lines w.r.t. each axis. The gray x's

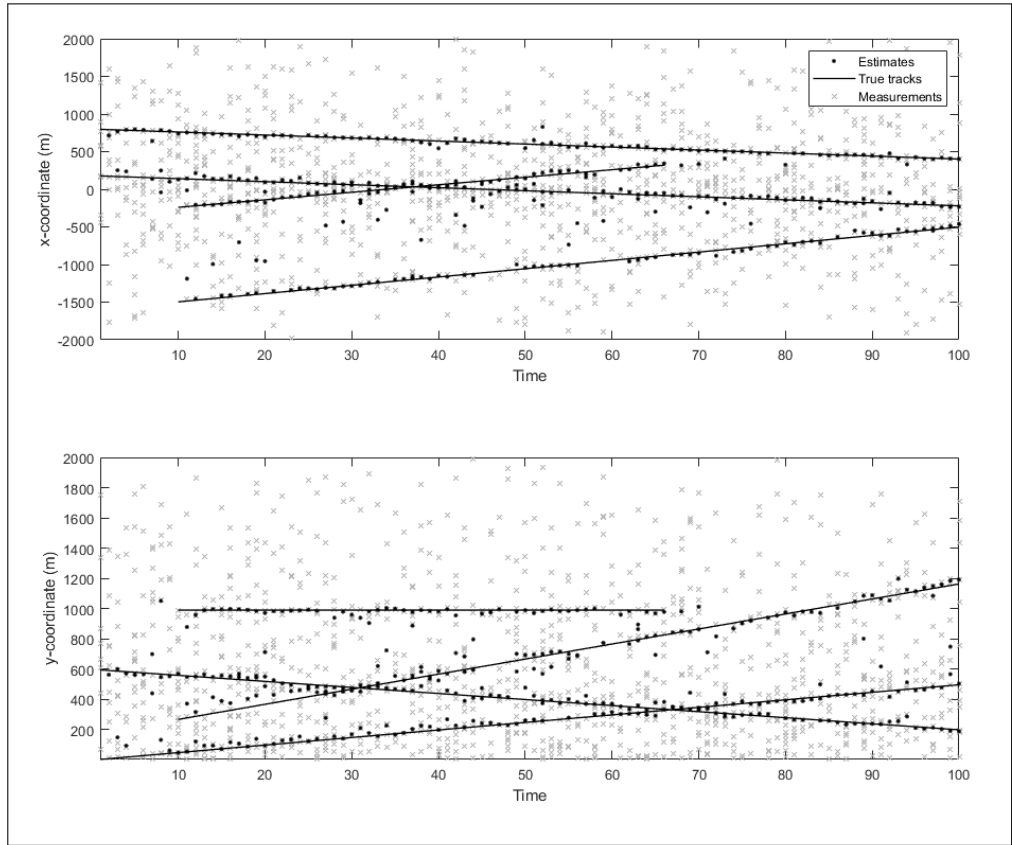


Figure 3.4: Measurements and estimates in each axis.

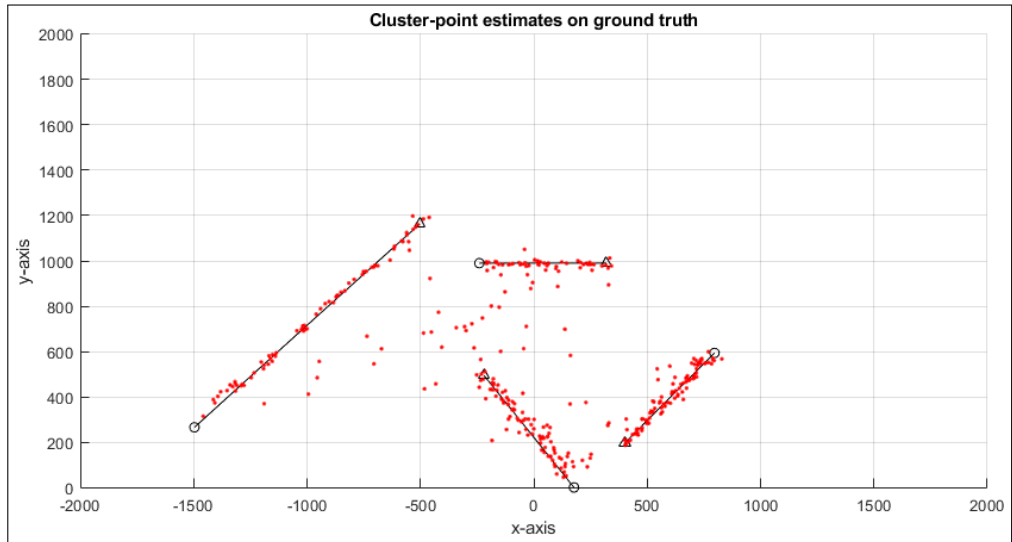


Figure 3.5: Estimates vs. ground truths.

indicate measurements collected at respective discrete time instants. It can be seen from the figure that the clutter density is higher in the scenario. Finally, the estimated track points are plotted in black dots.

It can be inferred from the figure that between the time intervals 35 – 50, the

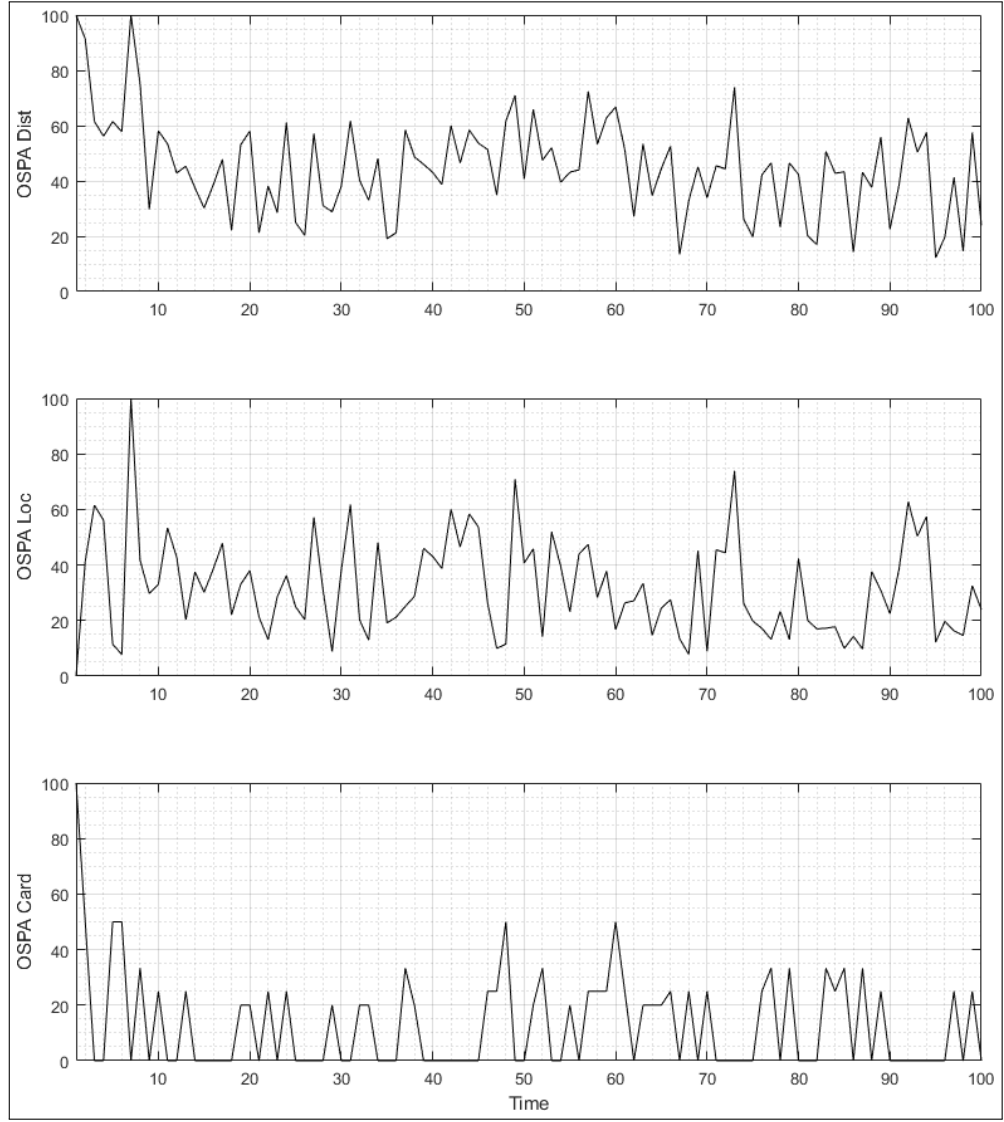


Figure 3.6: Performance metric OSPA cardinality, localization, and total errors for scenario-1.

two targets cross each other, or remain in very close proximity to each other. This is particularly a potential problem when the targets generate cluster measurements. It is quite possible for the filter to merge two targets and produce a single track.

The figure shows the qualitative performance of the proposed CT-PHD filter to be quite good. Except for a couple of points that can be seen in the y-coordinate plot in Fig.(3.8) as isolated points, the estimated track points are accurate, despite the dense clutter.

Fig. (3.9) shows the cardinality estimation results of the proposed CT-PHD filter. As can be seen from the figure, three targets start moving simultaneously and one target leaves the scene at time instant $k = 70$. From the figure, it is clear that the cardinality estimation performance is quite good. Again, as a result of the high variance in the

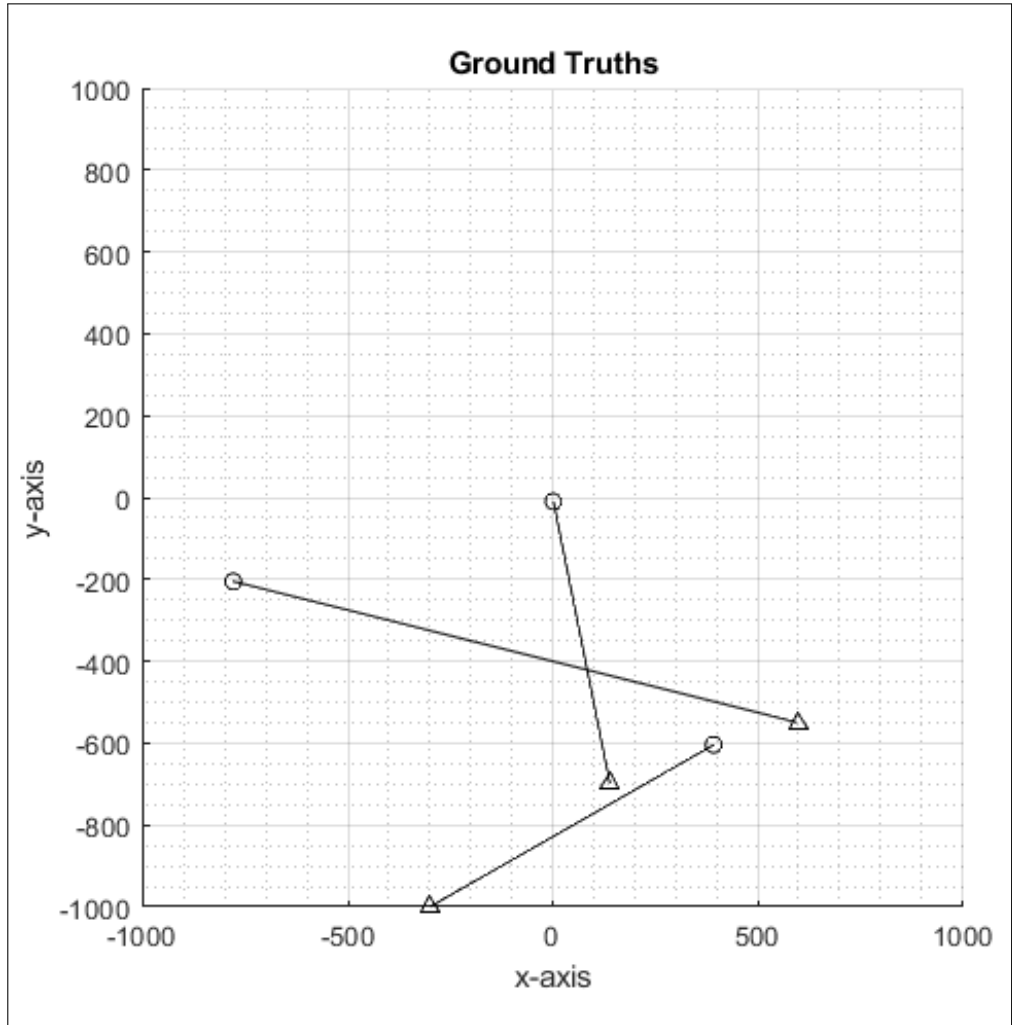


Figure 3.7: Ground truth trajectories for synthetic Scenario-2.

cardinality estimation of the PHD filter, some errors have been seen to occur.

In Fig. (3.10) the estimated track points are plotted in red on true target trajectories. Again, the qualitative performance of the proposed filter is apparent from the figure.

It should also be noted that since the proposed filter is based on the PHD filtering framework, the estimated track points have no connection in time (hence the *filter* in the name, rather than tracker). In other words, there is no systematic labeling of target tracks in the PHD concept, unless explicitly assigned. Thus, it is not abnormal to see some isolated track point estimates in the surveillance region as false tracks. Regardless, the PHD filter is a fundamental and widely accepted multi-target tracking filter.

Fig. (3.11) shows the quantitative performance of the CT-PHD filter in terms of OSPA errors. Notice, from the localization error plot, that the position estimation

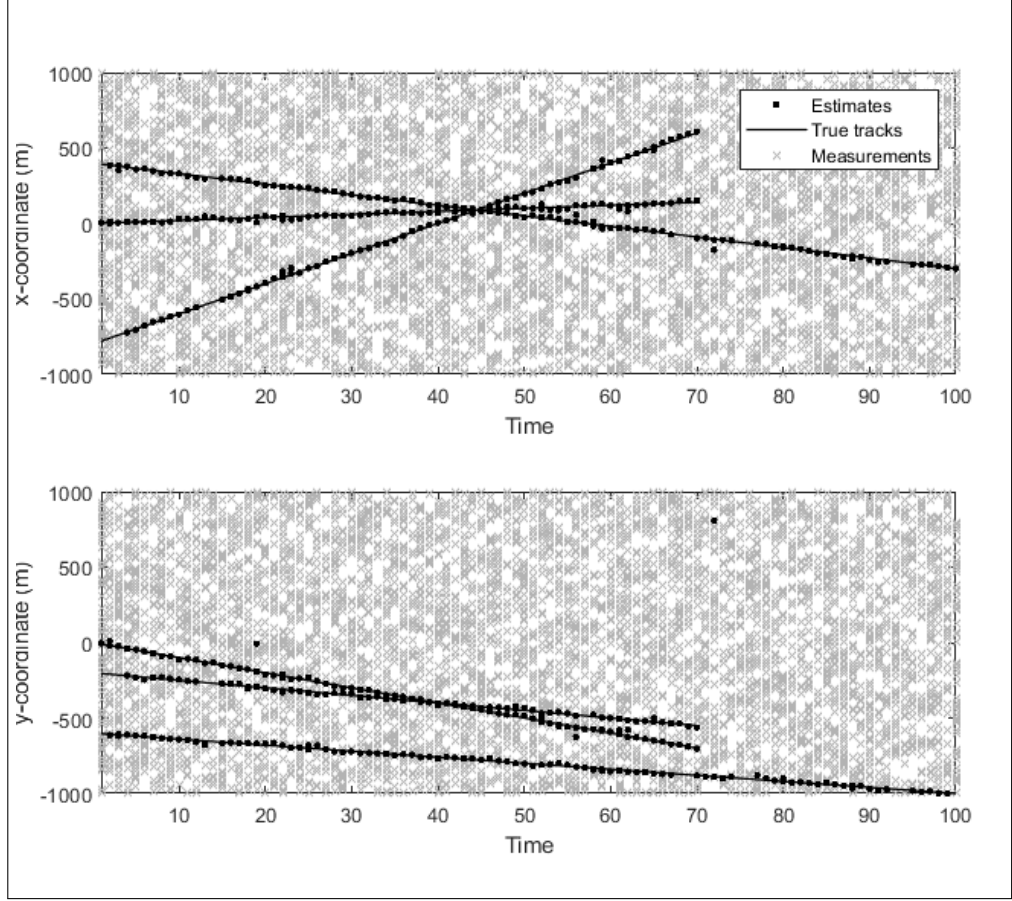


Figure 3.8: Estimated target trajectories and measurements w.r.t. each axes.

accuracy is around $10m$. Recall that target crossing occurs between 35 – 50'th time instants. From the OSPA cardinality error graph in Fig. (3.11), in this interval there is no notable cardinality error, thus showing the robustness of the proposed dynamic clustering method when there are closely spaced targets. The overall OSPA error is found to be around $40m$ during this scenario, which is arguably acceptable for a PHD-based filter.

In Fig. (3.12), a snapshot of the surveillance area is shown at time instant $k = 45$. In the figure, at the top right corner, a zoomed-in view of the area indicated in a red rectangle is also shown. The two measurement clusters of targets can be seen to be close to each other.

The synthetic scenarios show the effectiveness of the proposed CT-PHD filter in different situations. Simulation studies show that the filter exhibits promising performance for dynamic clustering.

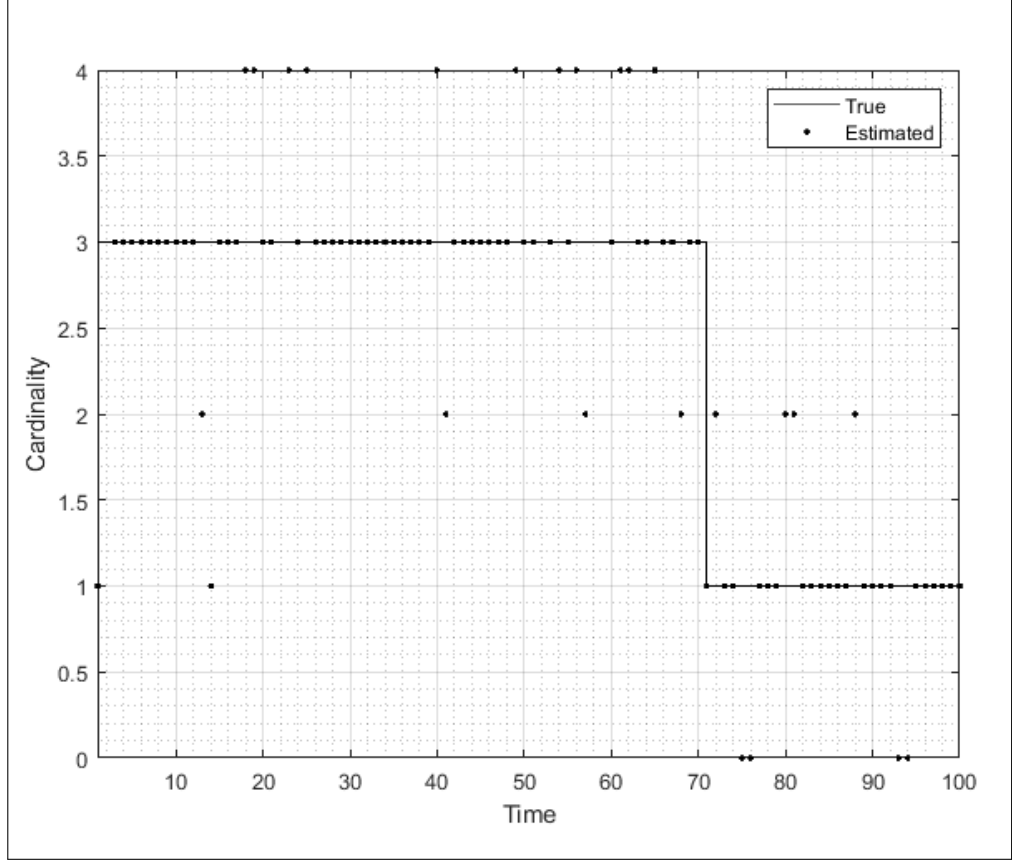


Figure 3.9: Estimated cardinality vs. ground truth cardinality for Scenario-2.

3.6.3. Real flight scenario

In this section, we present simulation results for the cluster-target tracking problem obtained by employing the algorithm delineated in the preceding section, utilizing real data. The data is derived from the actual flight trajectories of 10 targets. Measurements are acquired via the time difference of arrival (TDOA) method, using signals of opportunity emitted by the targets.

The received signal pulses, when detected by all four receivers, generate target detections accompanied by some random error. The tracking algorithm processes these detections. However, typically, communication signals yield a significantly larger number of pulses (and hence, detections) than the tracking algorithm can effectively manage in a physically meaningful manner. Consequently, the detections are buffered for a specific duration of time, and at the end of each time period, the target tracks are updated by the tracker. This buffering process naturally introduces a detection cloud for each time period, which, in this example, spans 3 seconds. The tracking algorithm must associate the detections with the target tracks. However, in this association

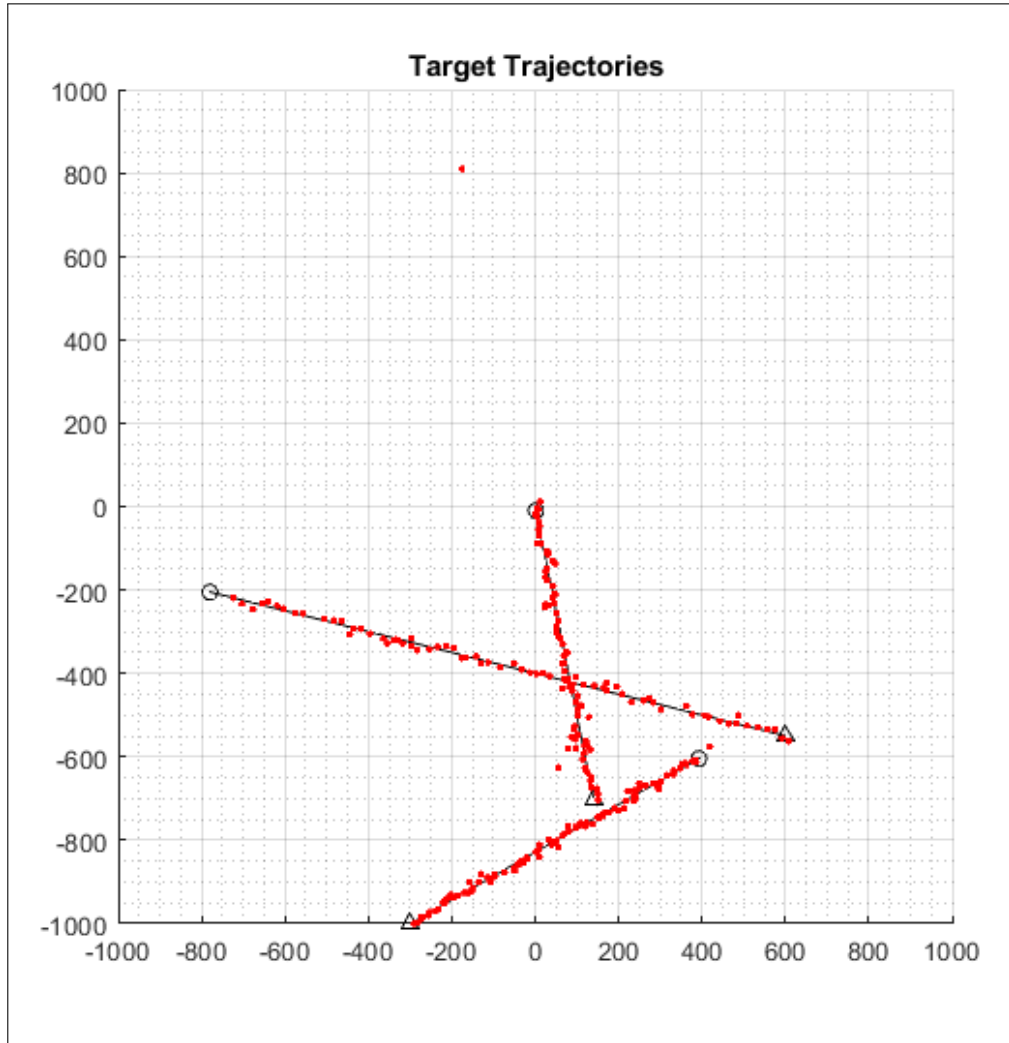


Figure 3.10: Estimated target trajectories for Scenario-2.

problem, multiple detections are linked to each target track, thereby transforming the multi-target tracking problem into a multi-cluster tracking problem.

In each scan, it is essential to initially identify each detection cluster and partition the measurement set into an indeterminate number of clusters. Owing to the unknown number of clusters (potentially including false detections) in each scan, the widely recognized $k - means$ clustering algorithm proves unsuitable, as it necessitates prior knowledge of the cluster count. To accomplish this partitioning, we employ the Dirichlet process mixture method, which is based on the Bayesian nonparametric framework. Subsequently, the statistics of the estimated clusters are utilized within the proposed multi-cluster tracking algorithm.

In Fig. (3.13), the target trajectories are shown with the target tags. The origin of the surveillance area is selected as the center of the receivers. The trajectories belong to the real targets which are obtained from real flights. It can be seen from

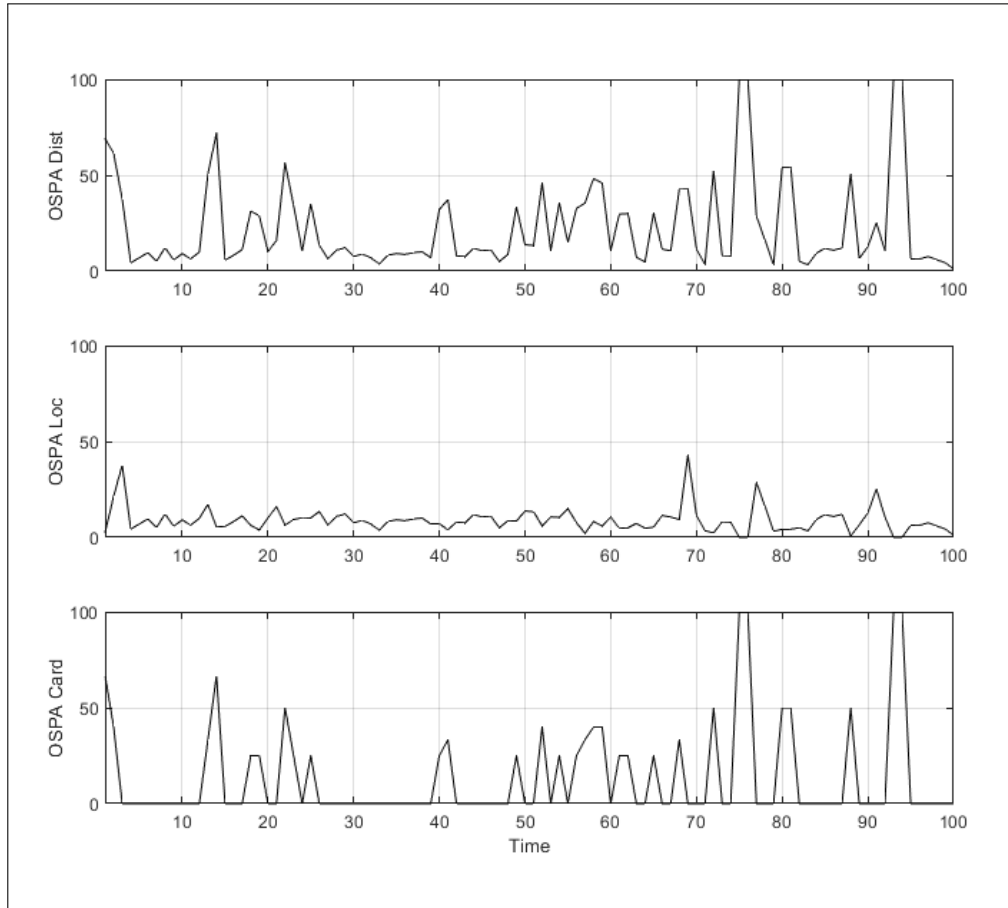


Figure 3.11: Performance metric OSPA total, localization, and cardinality errors.

the figure that while some targets perform straight flights, some targets perform maneuvers. Therefore, the motion model employed in the tracking algorithm is a non-linear coordinated turn model. The scenario is approximately 600 seconds in total.

In Fig. (3.14), the generated detections for a single scan of the scenario are illustrated. The red 'x' marks represent the detections, while the green circles indicate the ground truth positions of the targets. At each scan, each target yields a cluster of detections. The figure reveals that certain targets produce two distinct clusters. It is important to note that this case is applicable when the targets are situated at a considerable distance from the receivers. Consequently, it becomes evident that the target-receiver geometry influences the distribution of the measurement clusters. When targets are in closer proximity to the receiver sensors, the resulting measurement cluster becomes more concentrated. Conversely, when targets are farther from the receiver sensors, the detection clusters are less concentrated and may even be separated.

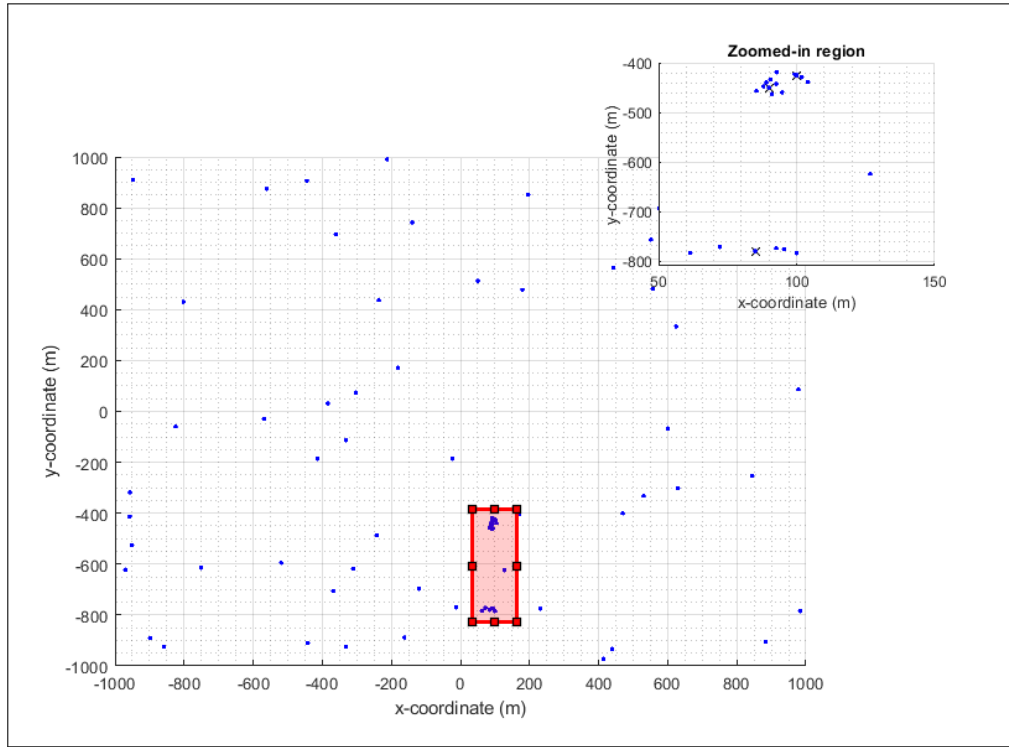


Figure 3.12: Measurement clusters at $k = 45$ s.

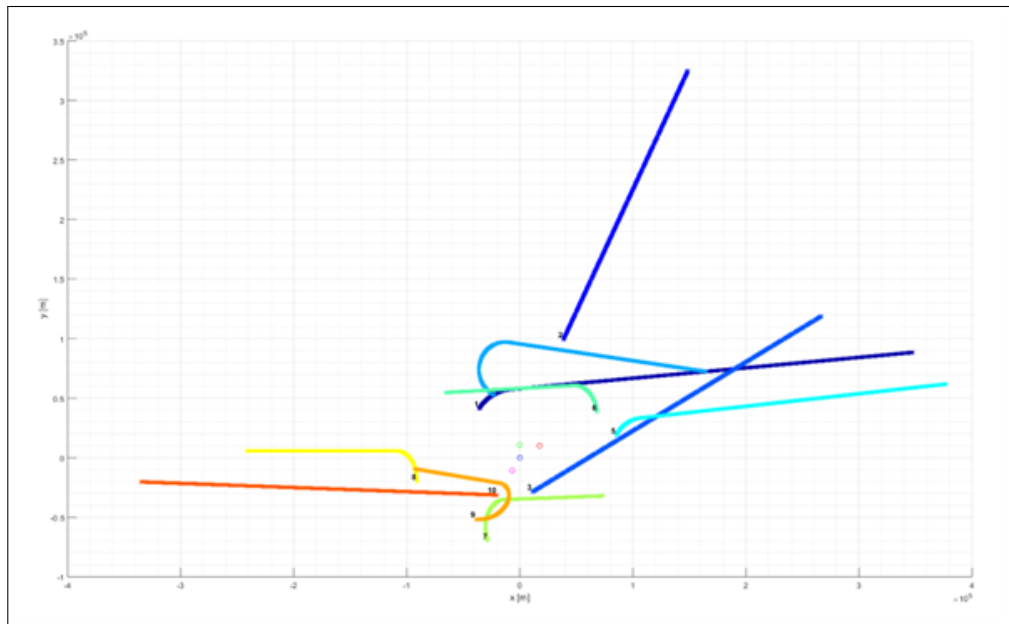


Figure 3.13: Ground-truth target trajectories and receiver positions.

Furthermore, besides the distance between the targets and the receivers, the targets' relative position also impacts the distribution of the clusters. As observed in the provided figure, certain clusters display scattering in a specific direction, whereas others exhibit a circular shape, indicating a lack of directionality. This phenomenon arises from the spatial arrangement of the receiver sensors. The geometry of the

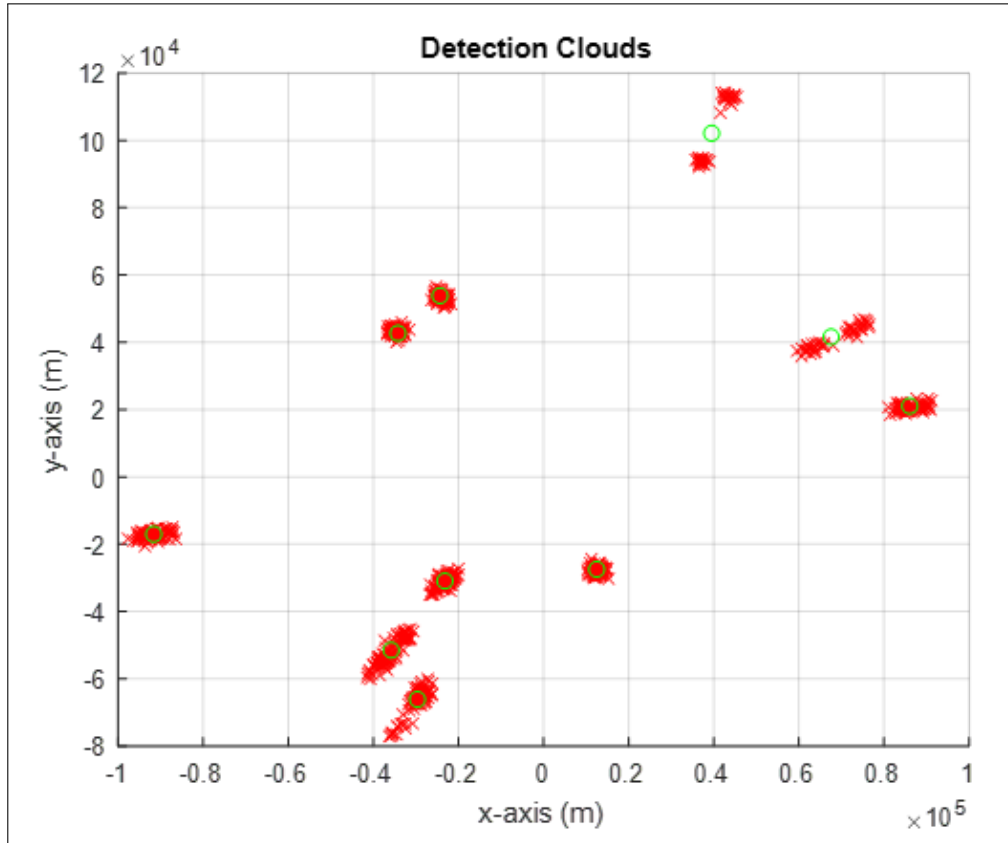


Figure 3.14: Measurement clouds and partitioned cluster centers.

receivers, in turn, influences the dispersion patterns of the measurement clusters. By examining these patterns, one can gain valuable insights into the relative positions of the targets and their effects on the distribution of the clusters, ultimately contributing to a more accurate tracking and analysis process.

The histogram plot displaying the number of detections generated during the same scan as Fig. (3.14) is depicted in Fig. (3.15). As observed in the figure, the number of generated detections can exhibit significant variation. The number of detections in each cluster can also change with the tracking period since the measurements are collected during *acquisition intervals*.

The tracking results for the given scenario are presented in Fig. (3.16). In this figure, the blue points represent the ground truth locations of the targets, while the green circles indicate the estimated positions derived from the tracking algorithm. It is evident from the figure that the algorithm encounters difficulties, particularly when the targets are in close proximity to one another. Furthermore, one of the targets never generates a track due to clustering errors. This observation suggests that incorporating a cluster model contingent upon the current target state may yield improvements. Given that the cluster shape and size depend on the target's relative position to the

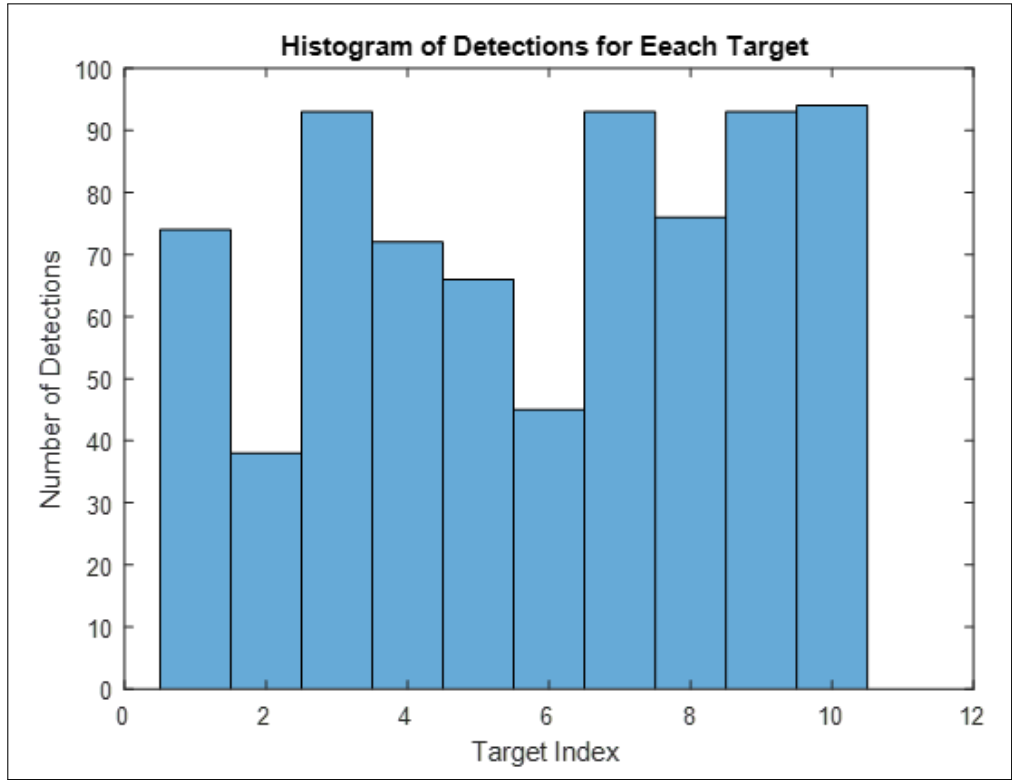


Figure 3.15: Histogram plot of the number of detections for each target in a single scan.

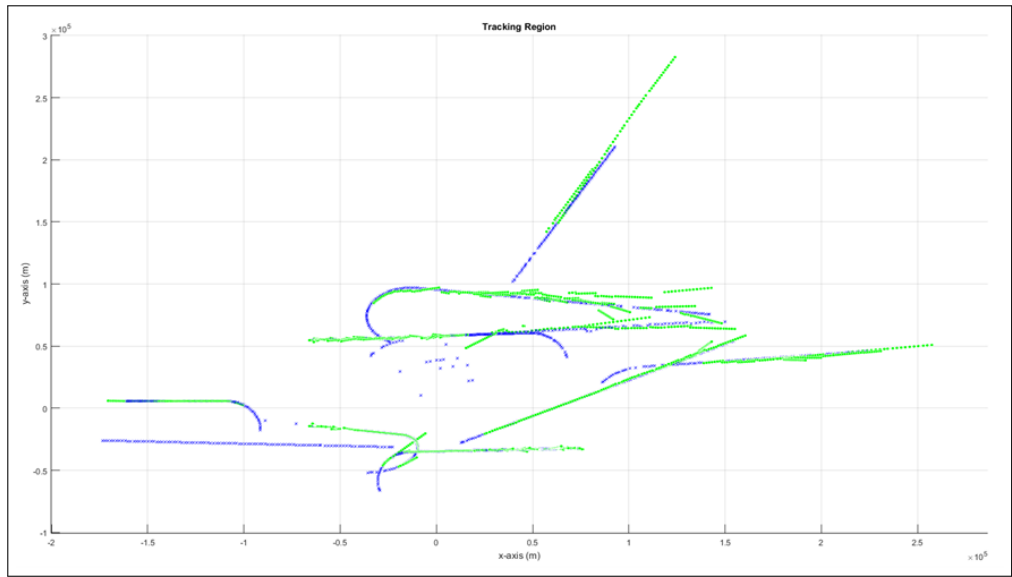


Figure 3.16: Tracking result of the 10-target scenario using the proposed algorithm.

sensors, introducing a state-dependent cluster shape and size model could potentially enhance the overall tracking performance.

The estimated number of targets by the tracking algorithm during the simulation is plotted in Fig. (3.17) against the time index. Because of the errors in the clustering

process, the tracking algorithm overestimates the number of targets. This is because some measurement clusters are split into two distinct clusters due to the target-sensor geometry. This can be overcome by employing target spawn models [33].

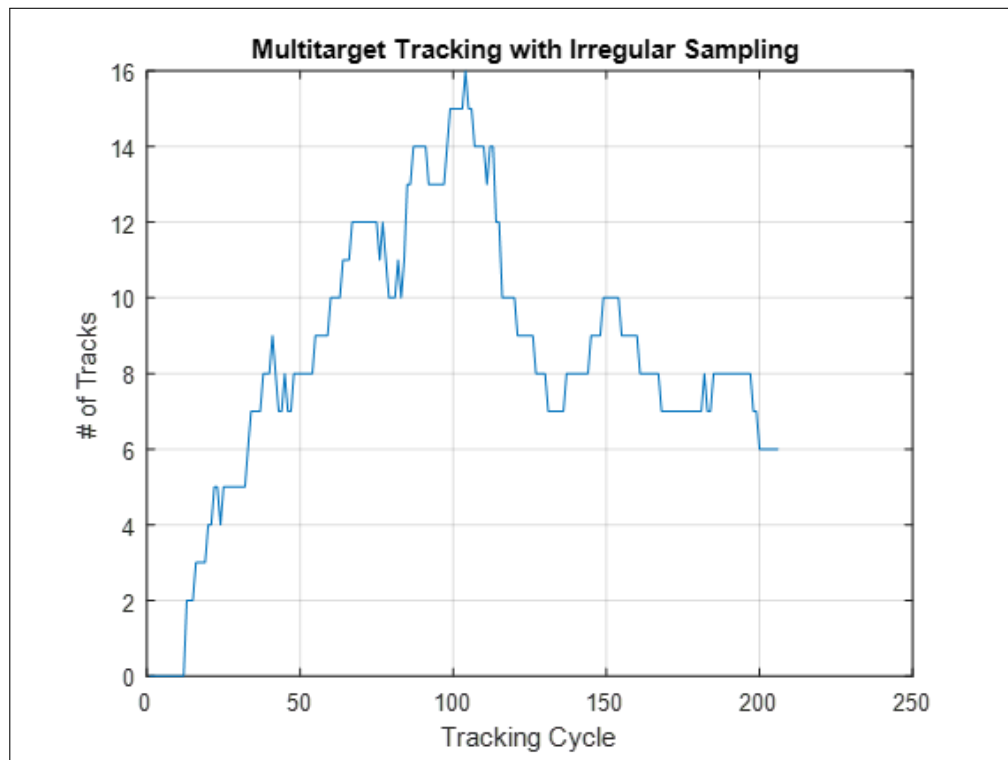


Figure 3.17: Estimated number of targets in the scenario.

4. LEVEL 4 FUSION: MULTI-TARGET SENSOR MANAGEMENT

In this chapter, the sensor management problem, which is categorized as level 4 fusion, is investigated. As mentioned before, the management process can improve the system's performance, as well as the reliability of the observation process. While the sensor management concept may be involved in a variety of applications, we approach the problem in the multi-target tracking concept. Ultimately, the goal is to improve the overall tracking performance or track quality.

We introduce the extended Hölder divergence (EHD) as a way to generalize the Cauchy-Schwarz divergence (CSD) for reward functions. The EHD is applied to a multi-target environment in RFS-based sensor management. The sensor management problem is approached as a partially observed Markov decision process (POMDP) [144]–[146], where the dynamics are modeled as a Markov process and only the posterior pdf of the true target state is known. Mathematical expressions for the EHD when the densities are Poisson RFSs, specifically for Poisson point processes with Gaussian mixture (GM) intensity functions, are derived and approximated in closed form. For more general models, numerical methods such as sequential Monte Carlo are used to calculate the EHD.

The chapter is organized as follows. Section 4.1. presents a brief introduction to sensor management. In Section 4.2., a literature review is given and the main concepts of sensor management are described, such as objective functions. Section 4.3. extends the Hölder divergence for the RFS densities. Section 4.4. describes the application of the proposed EHD to multi-target sensor management and numerical studies. Finally, Section 4.5. concludes the chapter.

4.1. Introduction

Sensor management is often referred to as the "Level 4 data fusion" process [33]. The performance of MS-MT systems is affected directly by the quality of the measured data about the environment, and sensor actions have a significant effect on measurement quality [147]. Therefore, multi-sensor management should be applied to multi-sensor systems to obtain the best possible measurements. For example, in a

bearings-only tracking scenario, the choice of ownship course change affects the target observability, directly affecting the tracking performance.

The sensor management process can be modeled as a Partially Observable Markov Decision Process (POMDP) [146], [148], [149]. In this model, it is essential to employ a proper objective/reward function. The objective or reward function quantifies the quality of action to be taken by the sensor at a later step. The objective function often employs some performance metric such as posterior expected error of cardinality and state (PEECS) [146], [150], the posterior expected number of targets [148], [151], and optimal subpattern assignment (OSPA) [152]. These metrics fall into the category of task-based criteria. The task-based criteria are often employed when there is a single-criterion system objective [153]. However, when there are multiple objectives that compete with each other, it is a challenging task to utilize the task-based approaches.

The information-theoretic approaches solve this problem. They quantify the overall information gain for a specific sensor action, thus they are suitable for multiple competing objectives. Among the information-theoretic divergences used in this problem are Kullback-Leibler divergence (KLD) [154], Renyi divergence [148], [155], and more recently the Cauchy-Schwarz divergence (CSD) [96], [147], [156]. The KLD and Renyi divergence suffer from the computational requirements. They often cannot be computed analytically and require approximations. In contrast, CSD can present a tractable closed-form solution [96]. For the MS-MT systems, prior to computing the objective function, the local posteriors must be fused into the global posterior. Therefore, information fusion is crucial for the multi-sensor control process.

4.2. Literature Review

The sensor management process takes into account different factors. For example, the sensor or platform capabilities, characteristics of the target or surveillance environment, tactical objectives, and operational priorities. For target tracking systems, sensors may take various actions that affect the overall system performance, usually through observation quality. These possible actions include changing the sensor position, orientation, or kinematics if the platform is moving, utilization of operating patterns or modes [96], etc. Some target tracking systems, such as unmanned vehicles, guided missiles, and autonomous platforms operate on quite

limited resources. The sensor management or control process, to be more specific, is especially important in such systems so that the limited resources are efficiently utilized to gain a comprehensive understanding of the underlying scenario or environment [157].

In the procedure, the sensor management schemes can be implemented in three different ways. The process can be bottom-up or top-down, i.e. rule-based or control-theoretic, respectively. Top-down systems can achieve near-optimal decisions and are more robust. Second, they can be single-step or multi-step. Multi-step procedures often provide accurate and smooth solutions because they consider a future time window, while single-step procedures are more adaptable to the change in target characteristics. Third, the sensor management is based on an objective function, and this objective function can be either information-theoretic or mission-oriented [33].

The mathematical formulation of the control problem is usually done under the partially observed Markov decision process (POMDP) approach for the Bayesian filtering [144], [158], [159]. In the POMDP, the dynamics of the target are modeled as a Markov process, however, only the posterior pdf is known while the true target state remains unknown [156]. At a given time instance, the measurement likelihood is a function of the target state and the control action. The sensor control action is expressed by a reward or cost function, depending on the optimization problem formulation. For instance, the objective can be minimizing the total estimation error, i.e. MMSE, or can be maximizing the probability of target detection, the number of well-resolved targets or targets of interest [33], [160].

Since it enables defining the problem in a formal mathematical manner, we mainly focus on the information-theoretic sensor control scheme. In this approach, the goal is to minimize or maximize some given global objective function. These objective functions are based on the definition of information that is defined in a rigorous manner. In this regard, one of the first used objective functions is based on Shannon entropy [161], [162]. A variety of information divergence methods are proposed for information-theoretic sensor control such as Kullback-Leibler cross-entropy [148], [153], [163], Renyi α -divergence family [155] and Csiszar divergence family [33]. More recently, Cauchy-Schwarz divergence is used as the objective function for its several theoretical advantages in the context of multi-target tracking [96], [164], [165].

The modeling of the sensor control scheme can be investigated in different configurations. For example, single-sensor single-target, single-sensor multi-target, and multi-sensor multi-target. Because of the different characteristics of the

above-mentioned configurations, they should be tackled in different ways. Although the ultimate aim of this work is to develop a multi-target sensor control scheme, for the sake of completeness, we will also discuss the other configurations throughout this work and present as detailed information as possible.

In this section, we will present a summary of the stochastic sensor control problem for target tracking. In the context of the target tracking framework, the stochastic control problem becomes especially challenging due to the high nonlinearity, complexity, and different uncertainties stochastic or otherwise [157].

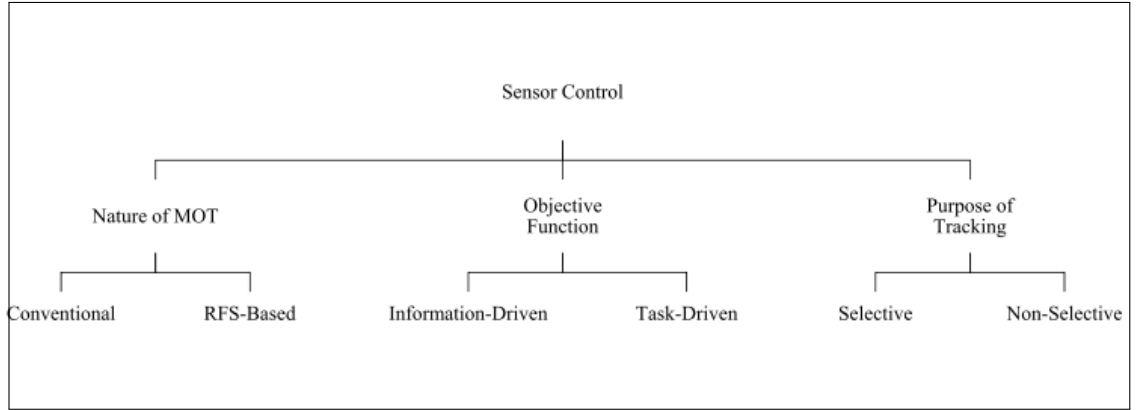


Figure 4.1: Taxonomy of sensor management/control approaches for MTT.

The 4.1 shows a more general classification of the solutions of the sensor control solutions for the target tracking. Regardless, we will mostly focus on the information-driven branch of the given classification tree.

Sensor control is a non-linear, stochastic control problem, involving many uncertainties about the state of the system, observations, etc. The control decision is made in the presence of these uncertainties. In the remainder of this section, we review the main modeling and objective function concepts of sensor management such as POMDP, and commonly applied task-driven approaches and Information-theoretic approaches.

4.2.1. Partially Observed Markov Decision Process (POMDP)

The POMDP model [157], [166] is a generalization of the Markov decision process. In the POMDP model, the sensor actions and system state are defined as Markov decision processes. However, the sensor only partially observes the

underlying system state, thus making it POMDP, and makes decisions under uncertainty. Then, when a new set of observations is available, it may update its belief of the true state, i.e. its posterior distribution. This results in the sensor taking action to improve its estimate of the system state. The Markov models and their relation to the state control are summarized in Figure 4.2.

Markov Models		Control over the state transition?	
		NO	YES
States are completely observable?	YES	Markov Chain	Markov Decision Process (MDP)
	NO	Hidden Markov Model (HMM)	Partially Observed Markov Decision Process (POMDP)

Figure 4.2: Markov models and their relation to state estimation.

4.2.2. Objective Functions

The objective functions are rooted in the center and are one of the core concepts of the sensor management process. They need to encapsulate the current situation and lead up to making optimal decisions to utilize the sensor resources appropriately.

The objective functions are split into two categories, in general, task-driven and information-theoretic. While task-driven methods, as indicated by the name, can be designed according to the mission requirements, often they are not mathematically rigorous. On the other hand, the information-theoretic methods provide the needed "theoretical basis" to define the information of the current situation mathematically and may lead to mathematically optimal decisions. However, they are usually not physically meaningful. Thus it is unclear how information-theoretic objective functions are related to mission-specific requirements, such as better localization of the targets being tracked.

In this section, we will discuss the commonly encountered objective functions, both task-driven and information-theoretic, in the target tracking literature.

4.2.2.1. Task-Driven Approach

Mahler pioneered task-driven methods in the early 2000s based on the RFS framework [33], [167]. He devised the mathematical foundations for sensor management approaches using the statistics of RFS densities. In task-driven methods, the objective function is usually constructed in the form of a cost function which is to be minimized. This section summarizes some of the widely used task-driven methods that can be found in the target tracking literature.

These methods converge on being heuristic approaches because their underlying logic is subjective and can be mission-specific. Since the definition of these methods usually depends on subjective factors such as commanding offices' or operators' decisions, designing a task-driven objective function to utilize these factors is, arguably, the main challenge [33]. Nevertheless, in the target tracking literature a handful of task-driven objective functions have been proposed and shown to be useful for certain cases. Some of these are summarized next.

4.2.2.1.1. Posterior Expected Number of Targets (PENT)

The posterior expected number of targets is proposed by Mahler [151] as an objective function for finite field-of-view sensors. The strategy selects the sensor action which will maximize the cardinality, or expected number of targets, of the posterior multi-target RFS density, hence the name PENT. Therefore, it can be considered a reward function. Mathematically, this reward function can be expressed as follows:

$$\mathcal{R}(u) = \mathbb{E}_{f_k(\cdot; u)}\{|\tilde{X}_k|\} = \int |\tilde{X}_k| \cdot p_k(X; u) \delta X. \quad (4.1)$$

(4.1), in fact, defines the expectation of the cardinality of RFS $\tilde{X}$ with respect to the posterior RFS density $p_k(\cdot)$. In other words, the integral represents the expected a posteriori (EAP) estimation of the cardinality statistics of the posterior density. If the EAP operation can be replaced with the maximum a posteriori (MAP) estimation, then

the control policy will turn into the following form:

$$u^* = \arg \max_{u \in \mathbb{U}} |\hat{X}_k(u)|. \quad (4.2)$$

Note that, this objective function exploits the structure of multi-object densities in the RFS framework. Stated differently, in order to use this objective function, there must be a ”*cardinality distribution*” component of the posterior density. Thus, it is suitable for RFS filters such as PHD or GLMB.

4.2.2.1.2. Cardinality Variance

While the PENT approach is suited for the limited FoV sensors, it actually focuses on the *mean* of the cardinality distribution. Alternatively, one can take the accuracy (or variance) of the cardinality estimate into account, instead of the mean [153]. In this approach, the objective function to be minimized is based on the variance of the cardinality estimate. Mathematically, the objective function can be formulated as follows:

$$\mathcal{R}(u) = \sigma_{f_k(\cdot; u)}^2\{|\tilde{X}_k|\} = \sum_{m=1}^M (m - \mathbb{E}_{f_k(\cdot; u)}\{|\tilde{X}_k|\})^2 \cdot \Pr(|\tilde{X}_k| = m). \quad (4.3)$$

Note that, the proposed objective function is applicable for the multi-Bernoulli filters as follows:

$$\mathcal{R}(u) = \sum_{i=1}^M \tilde{r}_k^i(u) [1 - \tilde{r}_k^i(u)]. \quad (4.4)$$

The described objective function can be estimated using maximum a posteriori (MAP) estimation for cardinality variance [168].

$$\mathcal{R}(u) = \sum_{i=1}^M \left(m - |\hat{X}|_{MAP} \right)^2 \cdot \Pr(|\tilde{X}|_k = m), \quad (4.5)$$

$$|\hat{X}|_{MAP} = \arg \max_{m=1:M} \Pr(|\tilde{X}_k| = m). \quad (4.6)$$

4.2.2.1.3. Posterior Expected Error of Cardinality and States (PEECS)

The described previous two methods focus only on cardinality statistics of the multi-target distribution to decide the control action to be taken. The posterior expected error of cardinality and states (PEECS), however, takes into account both cardinality and states, as a linear combination of respective normalized errors [56]. The objective function is defined as

$$\mathcal{R}(u) = \eta \xi_{|X|}^2(u) + (1 - \eta) \xi_X^2(u). \quad (4.7)$$

In (4.7), normalized variances of state estimate and cardinality are denoted as ξ_X^2 and $\xi_{|X|}^2$, respectively. Note that the subscripts indicate a multi-target state set X and its cardinality $|X|$. The weight η is a free parameter and can be chosen to emphasize the relative weight of each term in the objective function.

Given the multi-target filter being used, such as a multi-Bernoulli or CPHD, and the specific implementation method, such as SMC or GM, the described objective function can be calculated in different ways. An inclined reader may refer to [157] for details on how it is calculated for different cases.

4.2.2.2. Information-Theoretic Approach

The information theory is an abstract concept and it is usually very hard to establish a connection with physical intuition, especially in surveillance applications such as target tracking. In multi-target tracking applications, the sensor is, at least, expected to maximize the number of well-resolved. Information-theoretic objective functions enable us to approximate these physical objectives in a mathematically rigorous manner through information divergence functionals. However, there is criticism and uncertainty on how these information-theoretic functionals relate to physical reality. Regardless, using information theory in sensor control applications provides the necessary tools to implement these objectives mathematically and enables the design of approximate, even exact closed-form, solutions to the sensor control problem.

The widely used information-theoretic objective (or reward) functions are summarized in the following paragraphs.

4.2.2.2.1. Shannon Entropy

Shannon entropy appears in [169] as the earliest work for the information-theoretic sensor control in state estimation and target tracking. The entropy can be explicitly written as

$$H(X) = \mathbb{E}[I(X)] = \mathbb{E}[-\log P(X)]. \quad (4.8)$$

Where X is a discrete random variable and $P(X)$ is the probability mass function of X , and H represents the Shannon entropy. It is defined as the expected information content of random variable X , where $I(X)$ the information is itself a random variable, too. The entropy can also be defined between two random variables X and Y , as the conditional entropy as

$$H(X|Y) = - \sum_{i,j} p(x_i, y_i) \log \frac{p(x_i, y_i)}{p(y_i)}. \quad (4.9)$$

For the continuous random variables, the Shannon entropy becomes the differential entropy. This quantifies the information gain, measured from a reference density (relative entropy), and is utilized in the sensor control problem. In the state estimation, the information-theoretic reward function is calculated between the current information state, i.e. the predicted posterior density $p_{k|k-1}(X_k|Z_{1:k-1})$ and the future updated posterior density $p_{k|k}(X_k|Z_{1:k-1}, Z_k(u_k))$ where u_k is the sensor action at time index k .

4.2.2.2.2. Kullback-Leibler Divergence

Kullback-Leibler divergence (KLD) [170], which is also called relative entropy, is related to Shannon entropy. It is a special case of a wider class of statistical divergences, called f-divergences. The KLD is defined, between two probability distributions, as

$$D_{KL}(P||Q) = \int p(x) \log \frac{p(x)}{q(x)} dx. \quad (4.10)$$

Note that, from 4.10, the divergence is asymmetric, i.e. $D_{KL}(P||Q) \neq D_{KL}(Q||P)$. In the Bayesian inference, it is interpreted as information gained from the prior probability

Q to the posterior probability P . In other words, it is the amount of information lost when Q is used to approximate P . KLD can be evaluated analytically for the Gaussian probability densities.

4.2.2.2.3. Rènyi Divergence

Rènyi divergence or sometimes called α divergence [163] is used commonly in sensor control as the objective function. Calculated between a prior and a posterior distribution, it is used to measure the information gain of some process that makes the prior evolve to the posterior. The Rènyi divergence between the prior and posterior distributions is defined as follows:

$$\mathcal{R}_\alpha(P||Q) = \frac{1}{\alpha - 1} \log \left(\int p(x)^{1-\alpha} \cdot q(x)^\alpha dx \right). \quad (4.11)$$

Where α is a tuning parameter in the interval $[0, 1]$. Rènyi divergence approaches to KLD when $\alpha \rightarrow 1$. It emphasizes the tails of the two distributions involved in the operation. The Rènyi divergence is proven useful in many applications. However, the main drawback is that the divergence has no closed-form solution, except for some special cases. Thus, it is more computationally demanding. It is often used based on Monte Carlo methods.

The characteristics of the $\alpha = 1$ values, which enable the establishment of conditional information and mutual information in communication theory, could be crucial or irrelevant in different contexts, depending on the specific needs and objectives of each application.

The Rènyi divergence, essentially, measures the dissimilarity between two distributions. In the state estimation framework for sensor control, it can be interpreted as the information gained from the observations based on certain sensor actions [155].

4.2.2.2.4. Cauchy-Schwarz Divergence

Recently, the Cauchy-Schwarz divergence has attracted attention in the target tracking community. It has different mathematical and information-geometric properties. The Cauchy-Schwarz divergence (CSD) is defined as follows:

$$D_{CS}(p, q) = -\ln \frac{\langle p, q \rangle}{\|p\| \cdot \|q\|}. \quad (4.12)$$

Where $\langle p, q \rangle \triangleq \int p(x)q(x)dx$ is the inner product of two densities, and $\|p\| \triangleq \sqrt{\langle p, p \rangle}$. The CSD measures the distance between two densities [157], [171]. Note that, when $p = q$, the $D_{CS} = 0$, since the ratio inside the logarithm is unity. It is a positive and symmetric function such that $D_{CS}(p, q) = D_{CS}(q, p)$. See that the CSD is, geometrically, the cosine of the angle between two probability densities in the space of density functions. It can also be seen as an approximation to the KLD [157]. The CSD has other mathematical properties. It can be computed analytically for Gaussian mixture densities since it's the $L - 2$ distance between them [171]. In the concept of target tracking, it can also be evaluated in closed form between two PHDs that have Gaussian mixture intensity functions. It is also related to the Bhattacharyya distance between probability measures [171].

4.2.2.2.5. Hölder Divergence

Hölder divergence is a novel projective statistical divergence [172] which is built from Hölder's inequality. It is the generalization of the CSD. Since it is a projective divergence, it does not require the distributions to be normalized. In other words, it is invariant to the scaling of its arguments. Mathematically, $D_H(\lambda p : \lambda' q) = D_H(p : q)$, where $\lambda, \lambda' > 0$. Projective divergences are also referred to as *angular divergences* or *cosine divergences* because they do not depend on the mass of the density measures [172].

Hölder divergence between two distributions is given as follows [172]:

$$D_{\alpha, \gamma}(p : q) = -\log \left(\frac{\int p(x)^{\gamma/\alpha} q(x)^{\gamma/\beta} dx}{\left(\int p(x)^{\gamma} dx \right)^{1/\alpha} \left(\int q(x)^{\gamma} dx \right)^{1/\beta}} \right). \quad (4.13)$$

where the parameters α and β are called conjugate exponents and satisfies $1/\alpha + 1/\beta = 1$ relationship, and γ is a free parameter. Given the conjugate exponents, α and β , and $\gamma > 0$, the Hölder divergence outlined in (4.13) is a proper divergence for probability measures and a pseudo-divergence for positive measures.

Note that, for $\alpha = \beta = \gamma = 2$, (4.13) becomes equivalent to CSD given in (4.12). Therefore, the CSD becomes a special case of the more general Hölder divergence. Also, when $\gamma = 1$ we get the skew Bhattacharyya distance [172].

A disadvantage of the Hölder divergence is that except for the distributions that belong to the exponential family (i.e. Gaussian distributions), there is no closed-form solution to (4.13). For the Gaussian distributions of the form $p(x) = \mathcal{N}(\mu_p, \Sigma_p)$ and $q(x) = \mathcal{N}(\mu_q, \Sigma_q)$, the closed-form expression of (4.13) is found as follows:

$$D_{\alpha, \gamma}(p : q) = \frac{1}{2\alpha} \log \left| \frac{\Sigma_p}{\gamma} \right| + \frac{\gamma}{2\alpha} \mu_p^T \Sigma_p^{-1} \mu_p + \frac{1}{2\beta} \log \left| \frac{\Sigma_q}{\gamma} \right| + \frac{\gamma}{2\beta} \mu_q^T \Sigma_q^{-1} \mu_q + \frac{1}{2} \log \left| \frac{\gamma}{\alpha} \Sigma_p^{-1} + \frac{\gamma}{\beta} \Sigma_q^{-1} \right|. \quad (4.14)$$

When the densities are not Gaussian (not members of the same exponential family, in general), mixture densities for example, the Hölder divergence can be approximated or estimated by Monte Carlo techniques.

4.3. Extended Hölder Divergence (EHD) for PPP

The modern multi-target filters assume more complex distribution forms than the Gaussian. In recent years, there has been considerable development in these filters. These filters, using the RFS framework and Mahler's FISST densities, outperform traditional approaches in many areas. Thus, motivated by this, we will extend the Hölder divergence to the RFSs for multi-target sensor control applications. Then, propose a sensor control algorithm based on the extended Hölder divergence.

We consider a widely used RFS multi-target tracking filter, the PHD filter, for this purpose, which is a first-order moment approximation of the recursive multi-target Bayes filter. The PHD filter assumes that the multi-target RFS density is a Poisson RFS and is completely characterized by its intensity function, which is its first-order moment [125].

The integrals in (4.13) are inner products between two probability distributions. The denominator terms can be written as $\langle p^\gamma, 1 \rangle^{1/\alpha}$ and $\langle q^\gamma, 1 \rangle^{1/\beta}$. The numerator term, similarly, $\langle p^{\gamma/\alpha}, q^{\gamma/\beta} \rangle$.

The FISST densities of two Poisson point processes are defined as

$$p(X) = e^{-\langle u, 1 \rangle} [Ku]^X \quad (4.15)$$

$$q(X) = e^{-\langle v, 1 \rangle} [Kv]^X \quad (4.16)$$

where $u(x)$ and $v(x)$ are the intensity functions (or the PHDs) of these RFSs, K is the unit of measurement, and $[Ku]^X = \prod_{x \in X} Ku(x)$ is the standard multi-object exponentiation [156]. Thus, the inner product in the numerator term, in terms of Poisson point processes, is

$$\langle p^{\gamma/\alpha}, q^{\gamma/\beta} \rangle = \left[(e^{-\langle u, 1 \rangle})^{1/\alpha} (e^{-\langle v, 1 \rangle})^{1/\beta} \right] \exp(K^{\gamma-1} \langle u^{\gamma/\alpha}, v^{\gamma/\beta} \rangle). \quad (4.17)$$

Similarly, the denominator is found to be

$$\begin{aligned} \langle p^\gamma, 1 \rangle^{1/\alpha} &= \left[(e^{-\langle u, 1 \rangle})^\gamma \exp(K^{\gamma-1} \langle u^\gamma, 1 \rangle) \right]^{1/\alpha} \\ \langle q^\gamma, 1 \rangle^{1/\beta} &= \left[(e^{-\langle v, 1 \rangle})^\gamma \exp(K^{\gamma-1} \langle v^\gamma, 1 \rangle) \right]^{1/\beta} \end{aligned} \quad (4.18)$$

Then, by plugging (4.17), and (4.18) into the (4.13), the Hölder divergence between two Poisson point processes can be written as follows

$$D_{\alpha, \gamma}(p : q) = K^{\gamma-1} \left(-\langle u^{\gamma/\alpha}, v^{\gamma/\beta} \rangle + \frac{1}{\alpha} \langle u^\gamma, 1 \rangle + \frac{1}{\beta} \langle v^\gamma, 1 \rangle \right). \quad (4.19)$$

Note that if we set $\alpha = \beta = \gamma = 2$, then we obtain the CSD expression for the Poisson point processes [171].

The full derivation of (4.19) is as follows. The inner product between two Poisson point processes is defined by the following

$$\langle p^{\frac{\gamma}{\alpha}}, q^{\frac{\gamma}{\beta}} \rangle = \left\langle (e^{-\langle u, 1 \rangle} [Ku]^X)^{\frac{\gamma}{\alpha}}, (e^{-\langle v, 1 \rangle} [Kv]^X)^{\frac{\gamma}{\beta}} \right\rangle. \quad (4.20)$$

where the integral is Mahler's set integral [86]. By taking the constant exponential terms outside the inner product we obtain,

$$\langle p^{\frac{\gamma}{\alpha}}, q^{\frac{\gamma}{\beta}} \rangle = (e^{-\langle u, 1 \rangle})^{\frac{\gamma}{\alpha}} (e^{-\langle v, 1 \rangle})^{\frac{\gamma}{\beta}} \langle ([Ku]^X)^{\frac{\gamma}{\alpha}}, [Kv]^X)^{\frac{\gamma}{\beta}} \rangle. \quad (4.21)$$

The exponents inside the inner product can interchange, and by change of variable, we write $r(x) = [Ku]^{\gamma/\alpha}$ and $s(x) = [Kv]^{\gamma/\beta}$ in the inner product, thus

$$\langle p^{\frac{\gamma}{\alpha}}, q^{\frac{\gamma}{\beta}} \rangle = (e^{-\langle u, 1 \rangle})^{\frac{\gamma}{\alpha}} (e^{-\langle v, 1 \rangle})^{\frac{\gamma}{\beta}} \int_{\mathcal{X}} [r(x)s(x)]^X \mu(dX). \quad (4.22)$$

where μ is the dominating measure. Applying the set integral, (4.22), we get

$$\langle p^{\frac{\gamma}{\alpha}}, q^{\frac{\gamma}{\beta}} \rangle = (e^{-\langle u, 1 \rangle})^{\frac{\gamma}{\alpha}} (e^{-\langle v, 1 \rangle})^{\frac{\gamma}{\beta}} \sum_{i=0}^{\infty} \frac{1}{i! K^i} \underbrace{\left[\int_{\mathcal{X}} r(x) s(x) dx \right]^i}_{\langle r, s \rangle^i}. \quad (4.23)$$

Then, the argument of the summation operator is recognized as a power series and can be rewritten as

$$\begin{aligned} \langle p^{\frac{\gamma}{\alpha}}, q^{\frac{\gamma}{\beta}} \rangle &= (e^{-\langle u, 1 \rangle})^{\frac{\gamma}{\alpha}} (e^{-\langle v, 1 \rangle})^{\frac{\gamma}{\beta}} \sum_{i=0}^{\infty} \frac{\langle r, s \rangle^i}{i! K^i} \\ &= (e^{-\langle u, 1 \rangle})^{\frac{\gamma}{\alpha}} (e^{-\langle v, 1 \rangle})^{\frac{\gamma}{\beta}} \exp(K^{-1} \langle r, s \rangle). \end{aligned} \quad (4.24)$$

Then, rewriting for $r(x)$ and $s(x)$,

$$\langle p^{\frac{\gamma}{\alpha}}, q^{\frac{\gamma}{\beta}} \rangle = (e^{-\langle u, 1 \rangle})^{\frac{\gamma}{\alpha}} (e^{-\langle v, 1 \rangle})^{\frac{\gamma}{\beta}} \exp\left(K^{-1} K^{\frac{\gamma}{\alpha}} K^{\frac{\gamma}{\beta}} \langle u^{\frac{\gamma}{\alpha}}, v^{\frac{\gamma}{\beta}} \rangle\right). \quad (4.25)$$

Recall that the Hölder exponents α and β satisfy the relationship $\frac{1}{\alpha} + \frac{1}{\beta} = 1$. Thus, the inner product of two Poisson point processes, raised to some arbitrary power, can be reduced to the inner product of their corresponding intensity functions, raised to the same power. The same methodology can be applied to find inner products in (4.19). Then, the Hölder divergence formula can be written as

$$D_{\alpha, \gamma}(p : q) = -\log \left(\frac{\exp\left(K^{\gamma-1} \langle u^{\frac{\gamma}{\alpha}}, v^{\frac{\gamma}{\beta}} \rangle\right)}{[\exp(K^{\gamma-1} \langle u^{\gamma}, 1 \rangle)]^{\frac{1}{\alpha}} [\exp(K^{\gamma-1} \langle v^{\gamma}, 1 \rangle)]^{\frac{1}{\beta}}} \right). \quad (4.26)$$

Note that, the constant exponential terms are common in both numerator and denominator, so they cancel each other out. By separating the logarithms in (4.26) and simplifying the exponential terms we finally obtain

$$D_{\alpha, \gamma}(p : q) = K^{\gamma-1} \left(-\langle u^{\frac{\gamma}{\alpha}}, v^{\frac{\gamma}{\beta}} \rangle + \frac{1}{\alpha} \langle u^{\gamma}, 1 \rangle + \frac{1}{\beta} \langle v^{\gamma}, 1 \rangle \right).$$

It is inferred from (4.19) that the HD measure between two Poisson RFSs can be described in terms of the respective intensity functions. Thus, how this EHD can be implemented boils down to how the intensity functions are modeled. For instance, if the intensity functions of the two Poisson RFSs are modeled as Gaussian densities, then (4.19) can be calculated in closed form as given in (4.14). Other modeling assumptions require special care.

4.3.1. Implementation of EHD for GM Models

The Gaussian intensity function assumption of the Poisson RFSs is not applicable to most problems. Instead, a GMM is usually used to implement the intensity functions for multi-target tracking problems. Since there is no closed-form expression of HD for statistical mixture models, in order to use the proposed EHD with the GMM-PHD filter implementations, we need to be able to approximate the inner product expressions in (4.19).

Assume that the intensity functions $u(x)$ and $v(x)$ in (4.19) are modeled as Gaussian mixtures as follows:

$$u(x) = \sum_{i=1}^N \omega_i \mathcal{N}(x; \mu_i, \Sigma_i), \quad (4.27)$$

$$v(x) = \sum_{j=1}^M \omega_j \mathcal{N}(x; \mu_j, \Sigma_j). \quad (4.28)$$

To formulate the closed-form expression for the Expected Hölder Divergence (EHD) when the intensity functions represent Gaussian mixtures, we necessitate the subsequent two lemmas:

Lemma 1. The noninteger power of a Gaussian mixture can be approximated by a first-order power series.

$$\left[\sum_{i=1}^N \omega_i \mathcal{N}(x; \mu_i, \Sigma_i) \right]^v \approx \sum_{i=1}^N \omega_i^v \mathcal{N}^v(x; \mu_i, \Sigma_i). \quad (4.29)$$

Proof. For the proof, see [111],[173]. □

Lemma 2. The power of a Gaussian density is still a (unnormalized) Gaussian with the same mean and scaled covariance

$$\mathcal{N}^v(x; \mu, \Sigma) = c(v, \Sigma) \mathcal{N}(x; \mu, \frac{\Sigma}{v}) \quad (4.30)$$

where $c(v, \Sigma) \triangleq \sqrt{\frac{|2\pi \frac{\Sigma}{v}|}{|2\pi \Sigma|^v}}$ is a scalar, independent of mean, μ .

Proof. For the proof, see [111]. □

Theorem 4.1. Then, the inner products in (4.19) can be written as follows:

$$\langle u^\gamma, 1 \rangle = \sum_{i=1}^N \omega_i^\gamma c(\gamma, \Sigma_i), \quad (4.31)$$

$$\langle v^\gamma, 1 \rangle = \sum_{j=1}^M \omega_j^\gamma c(\gamma, \Sigma_j), \quad (4.32)$$

$$\langle u^{\frac{\gamma}{\alpha}}, v^{\frac{\gamma}{\beta}} \rangle = \sum_{i=1}^N \sum_{j=1}^M w_i^{\frac{\gamma}{\alpha}} w_j^{\frac{\gamma}{\beta}} c\left(\frac{\gamma}{\alpha}, \Sigma_i\right) c\left(\frac{\gamma}{\beta}, \Sigma_j\right) \mathcal{N}\left(\mu_i; \mu_j, \frac{\alpha \Sigma_i + \beta \Sigma_j}{\gamma}\right). \quad (4.33)$$

Proof. Proof of the results (4.31,-4.33) is given in Appendix 4.6.1.. $\square$

Based on the result of Theorem 4.1, the closed-form expression for the extended Hölder divergence when the underlying intensity functions are Gaussian mixtures can be written as follows:

$$D_{\alpha, \gamma}(p : q) = K^{\gamma-1} \left[- \sum_{i=1}^N \sum_{j=1}^M w_i^{\frac{\gamma}{\alpha}} w_j^{\frac{\gamma}{\beta}} c\left(\frac{\gamma}{\alpha}, \Sigma_i\right) c\left(\frac{\gamma}{\beta}, \Sigma_j\right) \pi_{ij}\left(\frac{\gamma}{\alpha}, \frac{\gamma}{\beta}\right) + \frac{1}{\alpha} \sum_{i=1}^N \omega_i^\gamma c(\gamma, \Sigma_i) + \frac{1}{\beta} \sum_{j=1}^M \omega_j^\gamma c(\gamma, \Sigma_j) \right]. \quad (4.34)$$

where $\pi_{ij}\left(\frac{\gamma}{\alpha}, \frac{\gamma}{\beta}\right) \triangleq \mathcal{N}\left(x_i - x_j; 0, \frac{\alpha P_i + \beta P_j}{\gamma}\right)$. Geometrically, the factor $\mathcal{N}\left(x_i - x_j; 0, \frac{\alpha P_i + \beta P_j}{\gamma}\right)^1$ measures the separation of the components i and j . Note that, the first-order power series approximation will be accurate when the mixture components are resolved sufficiently enough. Otherwise, before applying the approximation, some merging or pruning schemes should be applied to the close components and maintain the approximation accuracy. This merging can be done based on the Mahalanobis distance [58] or a similar metric, for example:

$$\sqrt{(x_i - x_j)^T \left(\frac{\alpha P_i + \beta P_j}{\gamma}\right)^{-1} (x_i - x_j)} \leq \eta \quad (4.35)$$

where η is a predetermined threshold. This ensures the well-separation of the Gaussian terms and increases the performance of the approximation strategy.

¹The $\mathcal{N}(\cdot)$ notation is used to emphasize the separation of the interpretation of the components.

4.3.2. Sequential Monte Carlo Implementation of EHD

For more general implementation schemes for the Poisson RFSs (or intensity functions of Poisson RFSs), well-known SMC or particle system implementations may be appropriate. With the increasing computational powers of computers, SMC implementations become widely accepted solutions to problems without a tractable analytic solution.

Theorem 4.2. Assume that the respective input distributions (intensity functions) of EHD in (4.19) are modeled as a set of weighted random particles, such as

$$u_{k+1|k}(x) = \sum_{i=1}^N w_{k+1|k}^{(i)} \delta_{x_{k+1|k}^{(i)}}(x), \quad (4.36)$$

$$v_{k+1|k+1}(x) = \sum_{i=1}^N w_{k+1|k+1}^{(i)} \delta_{x_{k+1|k+1}^{(i)}}(x) \quad (4.37)$$

where $u_{k+1|k}(x)$ is the intensity function of the predicted density and $v_{k+1|k+1}(x)$ is the intensity function of the hypothesized posterior density. Then, we can approximate the inner product integrals in (4.19).

$$\langle u_{\alpha}^{\gamma}, v_{\beta}^{\gamma} \rangle = \sum_{i=1}^N \left(\omega_{k+1|k}^{(i)} \right)^{\frac{1}{\alpha}} \left(\omega_{k+1|k+1}^{(i)} \right)^{\frac{1}{\beta}}, \quad (4.38)$$

$$\langle u^{\gamma}, 1 \rangle = \sum_{i=1}^N \left(\omega_{k+1|k}^{(i)} \right)^{\gamma}, \quad (4.39)$$

$$\langle v^{\gamma}, 1 \rangle = \sum_{i=1}^N \left(\omega_{k+1|k+1}^{(i)} \right)^{\gamma}. \quad (4.40)$$

Proof. The proof of (4.38)-(4.40) is presented in Appendix 4.6.2.. □

Note that, in (4.36) and (4.37) the support set of the particles for both predicted and posterior intensities are the same. This is because in particle filtering, only the weights are updated during the measurement update process, and particle sets remain the same [163], [174]².

Given (4.38)-(4.40), the EHD expression can be written as follows:

²For details on particle filtering see Ch. 2.1.3., Eq. (2.57)

$$D_{\alpha,\gamma}(p : q) = K^{\gamma-1} \left[- \sum_{i=1}^N \left(\omega_{k+1|k}^{(i)} \right)^{\frac{1}{\alpha}} \left(\omega_{k+1|k+1}^{(i)} \right)^{\frac{1}{\beta}} + \frac{1}{\alpha} \sum_{i=1}^N \left(\omega_{k+1|k}^{(i)} \right)^{\gamma} + \frac{1}{\beta} \sum_{i=1}^N \left(\omega_{k+1|k+1}^{(i)} \right)^{\gamma} \right]. \quad (4.41)$$

4.4. Application of EHD to Multi-Target Sensor Management

The sensor management problem is formulated as a *POMDP*, as typically done in the literature. The basic elements of a *POMDP* model are as follows:

- X_k : a set of object states at time instant k .
- $p_{k|k-1}(X_k|X_{k-1})$: transition density representing the time evolution of the system state.
- $\mathbb{U}$: set of admissible control actions
- Z_k : the set of observations at time instant k .
- $g_k(Z_k|X_k, u)$: a measurement model conditioned on the current state X_k and sensor control action $u_k \in \mathbb{U}$
- $\mathcal{R}_k(X_k, u_k)$: the objective function of system state X_k and sensor action u_k .

Thus, at any time instant k , the *POMDP* model can be represented as a 6-tuple, $\{X_k, p_{k|k-1}, \mathbb{U}, Z_k, g_k(Z_k|X_k, u), \mathcal{R}_k(X_k, u_k)\}$. Figure (4.3) illustrates the concept of sensor control in a multi-target tracking problem.

In general, the control action space is infinitely large [150]. Nonetheless, considering the infeasibility of accounting for an infinitely large space in practical applications, a predetermined finite set of acceptable control actions is employed within the framework of multi-target tracking. The *POMDP* model manages all forms of uncertainty uniformly and enables the execution of information-gathering actions.

In the *POMDP* model, the aim is to determine the control action $u^* \in \mathbb{U}$, which is a discrete action space, that optimizes an objective function. In the stochastic filtering context, this objective function is characterized by the posterior density of the system state and the candidate sensor control action. Mathematically, this is defined as follows.

$$u^* = \arg \max_{u \in \mathbb{U}} \mathbb{E}_{Z_k(u)} [\mathcal{R}_k(X_k, u_k)] \quad (4.42)$$

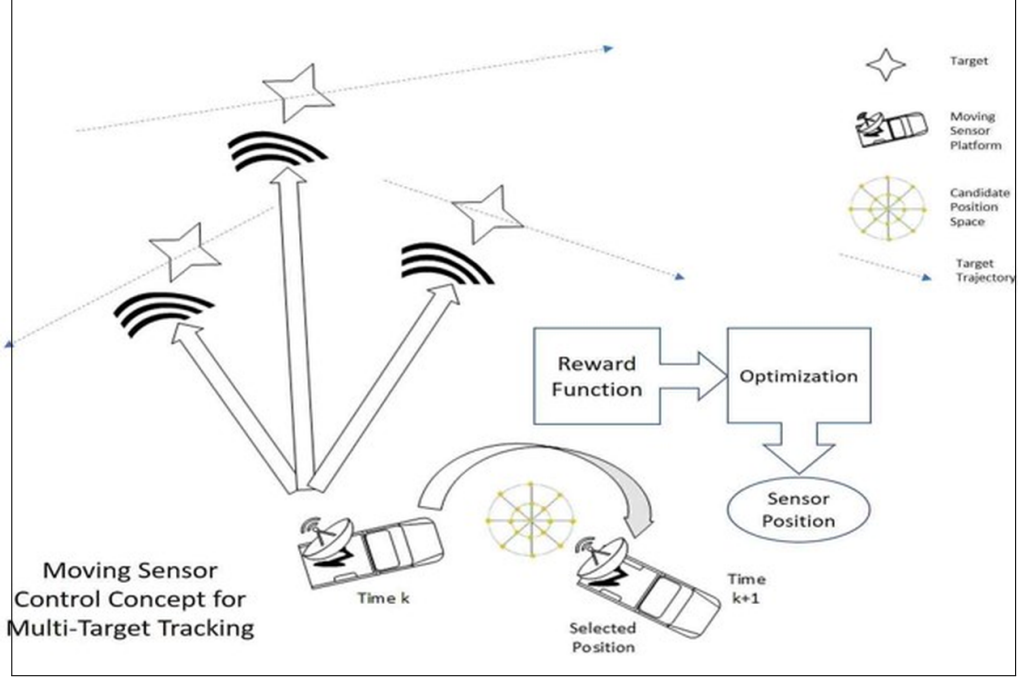


Figure 4.3: Illustration of moving sensor control concept for multi-target tracking.

According to the (4.42), the optimum sensor control action is found to be the one that maximizes the statistical expectation of the reward function $\mathcal{R}_k(X_k, u_k)$, w.r.t the current measurements. However, we have no knowledge of the posterior distribution at the time of sensor control. This is commonly solved by hypothesizing a set of ideal measurements for each sensor action. This process is called predicted ideal measurement set (PIMS) [150], [171], [175]. The *PIMS* is formed by generating noise-free measurements without any clutter and assuming unity detection probability, for each candidate sensor action in the action space.

$$Z_k(u) = \bigcup_{\hat{x} \in \hat{X}_{k|k-1}} \{h(\hat{x})\}. \quad (4.43)$$

A mathematical representation of the *PIMS* is shown in (4.43), where $h(\cdot)$ represents the measurement model. Then, the prior probability distribution of the system state is propagated using these hypothesized measurements. Based on the posterior distribution, the objective function is calculated and by maximizing its statistical expectation, the optimum sensor control action is chosen. Figure (4.4) shows the management algorithm.

In the figure, the following notation is used at the intermediary steps:

- $p_{k|k-1}(X_k|X_{k-1})$ is the predicted multi-target density from time $k-1$ to k ,

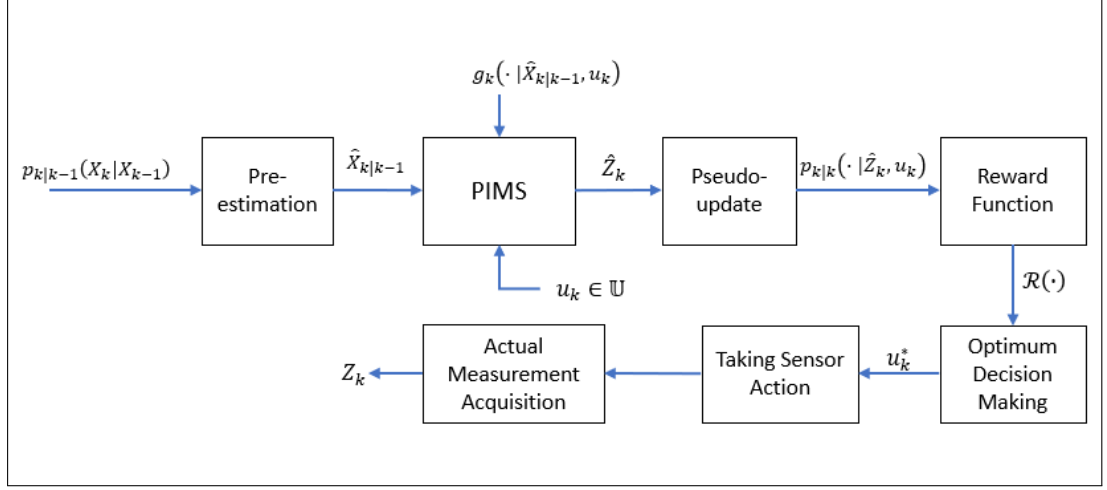


Figure 4.4: Flow chart of the sensor management scheme.

- $\hat{X}_{k|k-1}$ is the predicted target state set as a result of pre-estimation.
- $g_k(\cdot | \hat{X}_{k|k-1}, u_k)$ is the multi-target sensor likelihood function which uses the predicted target state set $\hat{X}_{k|k-1}$ and candidate control action u_k .
- $\hat{Z}_k$ is the predicted measurement set based on the $\hat{X}_{k|k-1}$ and u_k , obtained by applying *PIMS* (4.19).
- $p_{k|k}(\cdot | \hat{Z}_k, u_k)$ is the posterior calculated using predicted measurement set $\hat{Z}_k$.
- $\mathcal{R}(\cdot)$ is the reward value calculated using *EHD* between the predicted and posterior multi-target densities.
- u_k^* is the optimum control action based on the calculated reward function as shown in (4.18).
- Z_k is the actual measurement set obtained after carrying out the control action decided by the sensor management algorithm.

The reward function is based on the proposed *EHD* and the problem is formulated as an unconstrained optimization problem. Algorithm 1 describes the proposed sensor management algorithm based on the *EHD*.

4.4.1. Numerical Studies

The single object state is defined as $x = [x, y, v_x, v_y]^T$ which includes object position and velocity variables. The targets move according to the nearly constant velocity model. The transition density from time step $k-1$ to k is defined as a Gaussian $p_{k|k-1}(x_k|x_{k-1}) = \mathcal{N}(x_k; F_k x_{k-1}, Q_k)$ where F_k is the transition matrix and Q_k is the

Algorithm 1 : Information-theoretic sensor management subroutine.

Inputs: Time index k , predicted multi-target density $p_{k|k-1}(\cdot)$, measurement likelihood $g_k(\cdot|\hat{X}_{k|k-1}, u_k)$, control action set $\mathbb{U}$.

Output: Optimum sensor action u_k^* .

```

1:  $\hat{X}_{k|k-1} \leftarrow \text{Estimate}(p_{k|k-1}(\cdot))$  ▷ Estimate multi-target state
2:  $\hat{L}_{k|k-1} \leftarrow |\hat{X}_{k|k-1}|$  ▷ Estimate cardinality
3: for  $u \in \mathbb{U}$  do
4:    $\tilde{Z}_k \leftarrow \emptyset$  ▷ Initiate PIMS
5:   for  $j = 1 \dots \hat{L}_{k|k-1}$  do
6:      $\tilde{z} \leftarrow h(\hat{x}_j)$ 
7:      $\tilde{Z}_k \leftarrow \tilde{Z}_k \cup \{\tilde{z}\}$ 
8:      $p_{k|k}(\cdot|\tilde{Z}_k, u) \leftarrow \text{Update}(p_{k|k-1}(\cdot), g_k(\tilde{Z}_k(u)|\cdot))$ 
9:      $\mathcal{R}_k(u) \leftarrow D(p_{k|k-1}(\cdot), p_{k|k}(\cdot|\cdot, u))$  ▷ Compute reward
10:  $u_k^* \leftarrow \underset{u \in \mathbb{U}}{\text{argmax}} R(u)$ 
11: return  $u_k^*$  ▷ Optimum sensor action

```

process covariance, with $T = 1$. The total simulation time is $K = 100$ seconds.

The single target measurements consist of the range and bearing measurements and are represented as the vector $z = [\theta, r]^T$. The measurement model is represented mathematically as $z_k = h(x_k; \eta_k)$ where $\eta_k \sim \mathcal{N}(\cdot; 0, R_k)$ is the measurement noise with covariance matrix $R_k = \text{diag}(\sigma_\theta^2, \sigma_r^2)$. The measurement noise variances are calculated as follows

$$\sigma_r = \sigma_{r,0} + \beta_r \|p - s\|^2 \quad (4.44)$$

$$\sigma_\theta = \sigma_{\theta,0} + \beta_\theta \|p - s\|^2 \quad (4.45)$$

where the values of are $\sigma_{r,0} = 10m$, $\beta_r = 5 \times 10^{-5}m^{-1}$, $\sigma_{\theta,0} = \pi/180rad$, $\beta_\theta = 10^{-5}rad/m$, and p and s are the object position and sensor position, respectively. The sensor detection model, the probability of detection, is also defined adaptively [157].

$$p_D(s, p) = \begin{cases} 0.99, & \|p - s\| \leq R_0 \\ \max(0, 0.99 - h(\|p - s\| - R_0)), & \text{otherwise} \end{cases} \quad (4.46)$$

where $R_0 = 300m$ and $h = 5 \times 10^{-4}m^{-1}$. The sensor starting position is at the origin. At each time step, the sensor control algorithm determines the new sensor position, s_k , from the admissible sensor action set.

The parameters defining the action set are $N_R = 2$, $N_\theta = 8$, $\Delta R = 50m$ [176], which are the number of range steps, angle steps, and radial step size defining the action set, respectively. With these parameters, the admissible action set has 17 elements (including no move action). $\gamma = 2$ is borrowed from [172] as the free parameter for the EHD. Rényi-divergence parameter $\alpha = 0.5$ is used because it provides maximum discrimination when two densities are similar, as specified in [161], [176].

4.4.1.1. Scenario 1

There are 5 target trajectories for the pseudo-stationary target scenario. The described scenario is a widely used scenario for multi-target sensor control applications and in this paper used as a benchmark scenario [155], [163], [171]. The target velocities are very small, thus their positions are virtually constant. The scenario and algorithm parameters are given in the previous section. The number of targets in this scenario is not constant. 5 targets appear and disappear at different times in the scenario.

We compared the performance of the proposed reward function EHD against the Rényi [163] and CSD [171] and against the stationary sensor case, as a baseline. The comparison is made in terms of the OSPA metric.

Figure 4.5 shows the average total OSPA error of 100 Monte Carlo runs for each reward function and the stationary case. As can be seen from the figure, the proposed EHD-based reward function outperforms the other two methods, while the stationary sensor case is the worst one, as expected. The performance of Rényi and CSD-based reward functions are similar. Note that the standard PHD filter has a well-known cardinality bias [177], therefore the overall performance would be limited.

Another comparison between the three reward functions is made in terms of the average probability of detection. Since the $p_D(X_k, s_k)$ depends on the multi-object state, as stated in the (4.46), and the sensor control routines are expected to drive the platform towards maximizing the $p_D(X_k, s_k)$, it makes sense to compare the achieved probability of detection in time. Figure 4.6 shows the average probability of detection in time for each reward function. The probability of detection for each detected target is calculated at each time step as in (4.46). Then, in a Gaussian mixture setting, for each time step, we obtain the average probability of detection as in (4.47).

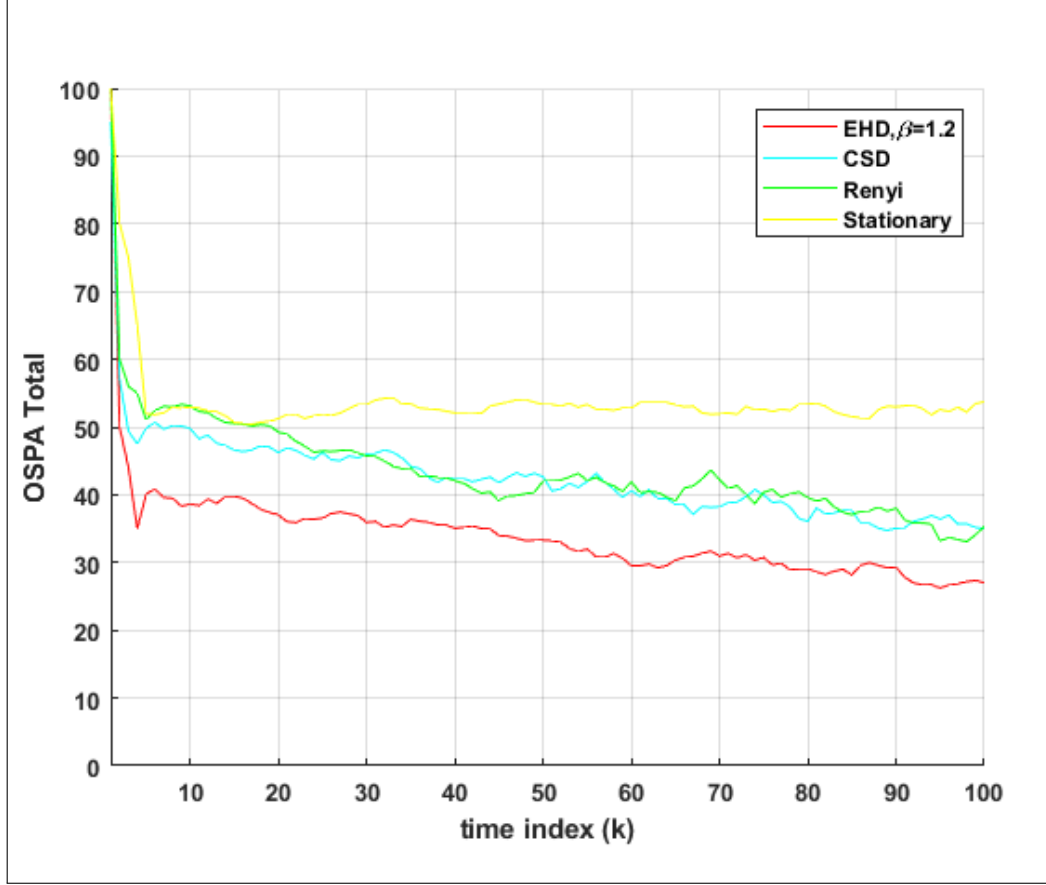


Figure 4.5: Total OSPA error for the three reward functions and the stationary sensor case.

$$\bar{p}_D(p_k, u_k) = \sum_{n=1}^{\hat{N}_k} \omega_{D,k}^n \mathcal{N}(p_{D,k}^{(n)}; s_k(u_k), P_{D,k}^{(n)}) \quad (4.47)$$

where k is the time index, $s_k(\cdot)$ is the sensor position, $p_{D,k}^{(n)}$ is the n th target track position at time k , $\hat{N}_k$ is the total number of tracks at time k , and u_k is the sensor control command at time k . In terms of probability of detection, as shown in the figure, CSD and EHD-based reward functions perform similarly, especially in the first half of the simulation. However, the performance of Rényi divergence decreases significantly between the time indices 70 – 80. The reason is that at $k = 70$, a target leaves the surveillance area, and it takes time for Rényi divergence to adjust to the new target geometry. However, EHD is not affected by the target death. This indicates that the proposed reward function is better at finding the optimum location and adjusting the change in the target geometry than the other two methods.

In this scenario, we showed that the proposed EHD outperforms the existing information-theoretic reward functions for the multi-target sensor control problem. In

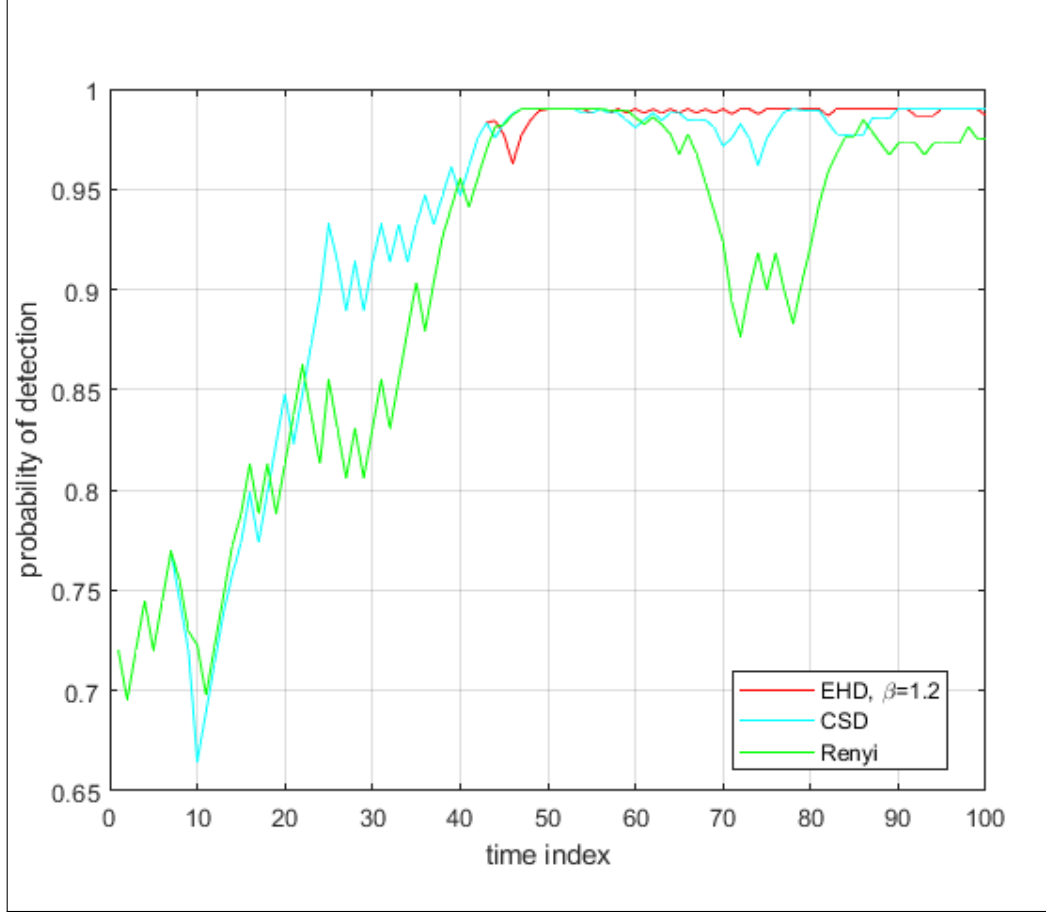


Figure 4.6: Average probability of detection profile for each reward function.

the total OSPA error, the superiority of the proposed EHD-based reward function is much more clear, while the Rényi and CSD-based reward functions have almost the same performance. Additionally, by comparing the probability of detection profiles, we confirmed that the sensor control function moves the sensor to better localize the targets.

4.4.1.2. Scenario 2

In this scenario, we investigated the effects of the parameters on the performance of the proposed EHD and Rényi reward functions. As can be seen from (4.19), it is possible to emphasize the effect of either prior or posterior in the divergence by choosing the exponents appropriately. For this purpose, we used $\beta = 1.2$ and $\beta = 10.2$ where the exponents satisfy the $1/\alpha + 1/\beta = 1$ relationship. We run Monte Carlo simulations for these two cases and calculate the OSPA values. The results are shown in Fig. 4.7 and 4.8.

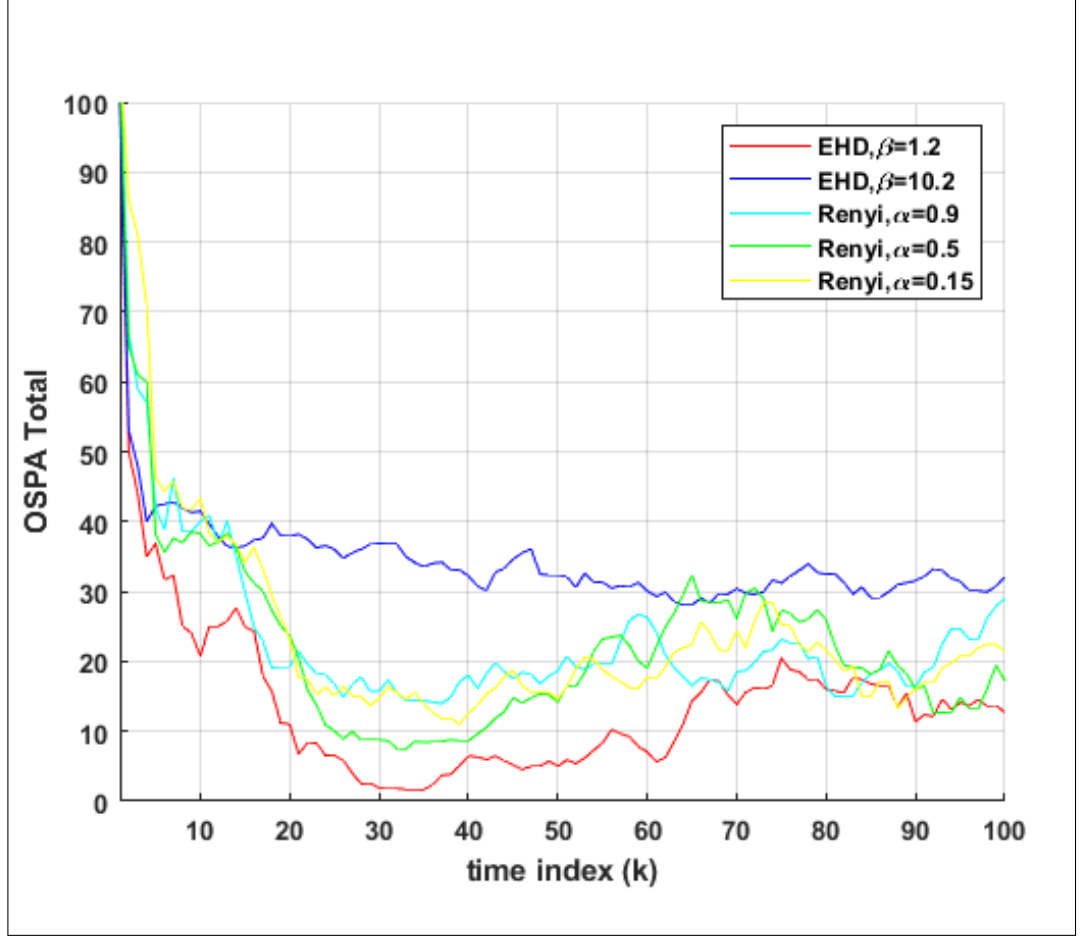


Figure 4.7: Total OSPA error, an average of 100 MC runs for EHD $\beta = \{1.2, 10.2\}$ and Rényi $\alpha = \{0.15, 0.5, 0.9\}$ for a dynamic environment.

In the first case, the targets appear and disappear in the scene at different times, as in Scenario-1. Between the time indices of 10 and 35, new targets are appearing in the area. It is observed that when β is close to 1, the proposed EHD function is more sensitive to appearing targets (i.e. change in the cardinality) in the scene. When β is small, the reward function captures the changes in the cardinality better and moves the sensor to localize the newly detected targets more effectively. Thus, it can be said that lower values of β are more suitable for dynamic environments where new targets are appearing/disappearing rapidly. While a similar statement can be made for the Rényi-based reward function, its performance is slightly worse than the proposed one. In the second case, the same targets appear in the scene at the beginning of the scenario and remain in the scene until the end. In this case, the total OSPA error is smaller when the parameter β is bigger, as shown in Figure 4.8. From this experiment, it is inferred that for static environments, the control algorithm can be tuned by setting the exponent to a large value.

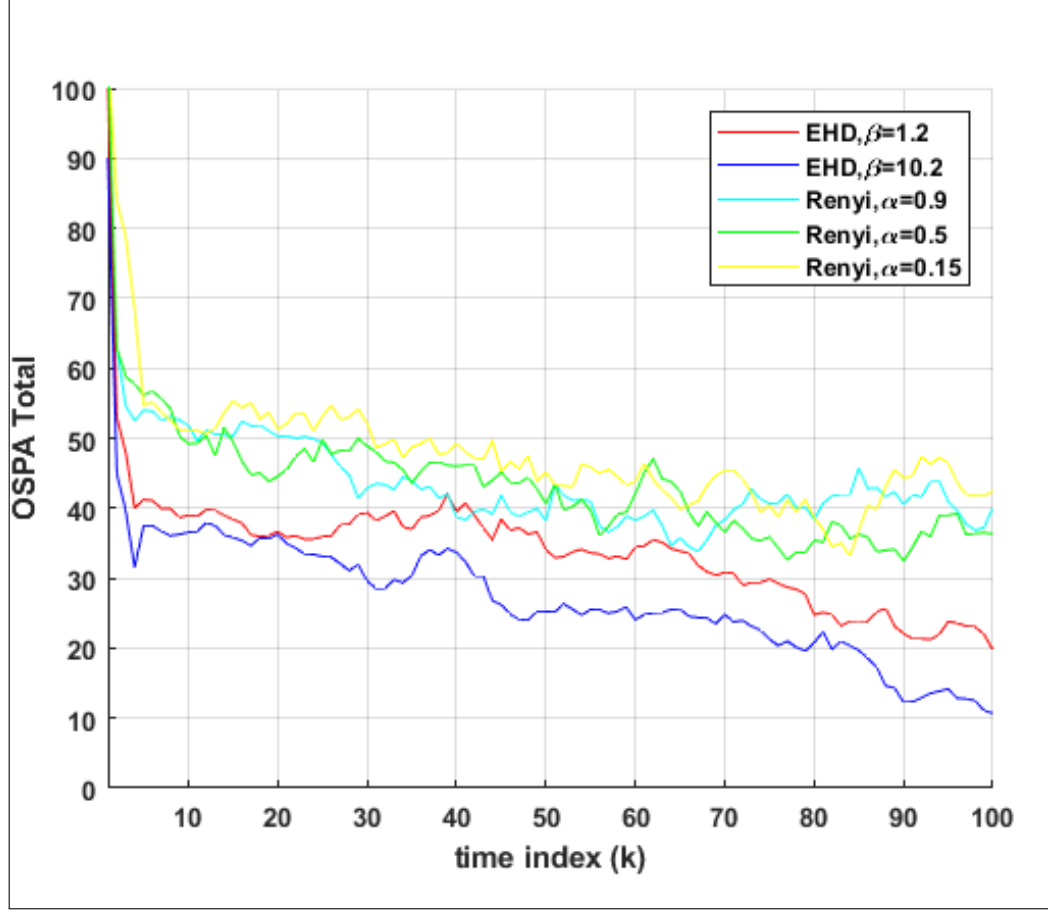


Figure 4.8: Total OSPA error, an average of 100 MC runs for EHD $\beta = \{1.2, 10.2\}$ and Rényi $\alpha = \{0.15, 0.5, 0.9\}$ for the static environment.

As can be seen from Figures 4.7 and 4.8, the effect of parameter change for the proposed EHD is more significant than Rényi in both static and dynamic cases. For Rényi, the sum of the exponents should be 1 (see (4.11)). In other words, the relation between the exponents is linear (i.e. $\alpha + \beta = 1$). However, since the Hölder exponents are conjugate, the relation is hyperbolic (i.e. $\frac{1}{\alpha} + \frac{1}{\beta} = 1$). Therefore, the change in one exponent for Rényi results in a linear change in the other, so the capability of emphasizing the information gain between the densities by parameter adjustment is limited. On the other hand, because of the hyperbolic relation between Hölder exponents, a small change in one parameter may result in a large change in the other. Thus, it can magnify the information gain between the densities significantly.

Also note that while the exponents in Rényi divergence can take values in the range $0 < \alpha < 1$, the Hölder exponents $\alpha, \beta > 1$. It is possible, by using EHD, to emphasize the information gain by when a specific sensor action is applied. Thus, in light of these observations, it can be said that the proposed EHD-based reward function is more capable of adapting to different situations via parameter adjustment than the

Rényi divergence-based method. Our results are aligned with the previous studies in [178]. In [53], [57], [179], the authors compared different information-theoretic divergences for sensor management.

For the same scenario, we compare the sensor trajectories suggested by the different reward functions. The obtained sensor trajectories are shown in Figure 4.9. From the figures, it can be inferred that the proposed EHD-based reward function results in a smoother sensor trajectory than CSD and Rényi-based ones. In contrast, the CSD and Rényi-based reward functions result in more exploratory behavior. We infer that this result stems from the reason explained in the previous paragraph. Thus, it can be said that in scenarios where the sensor is needed to move less, the EHD-based reward function performs better.

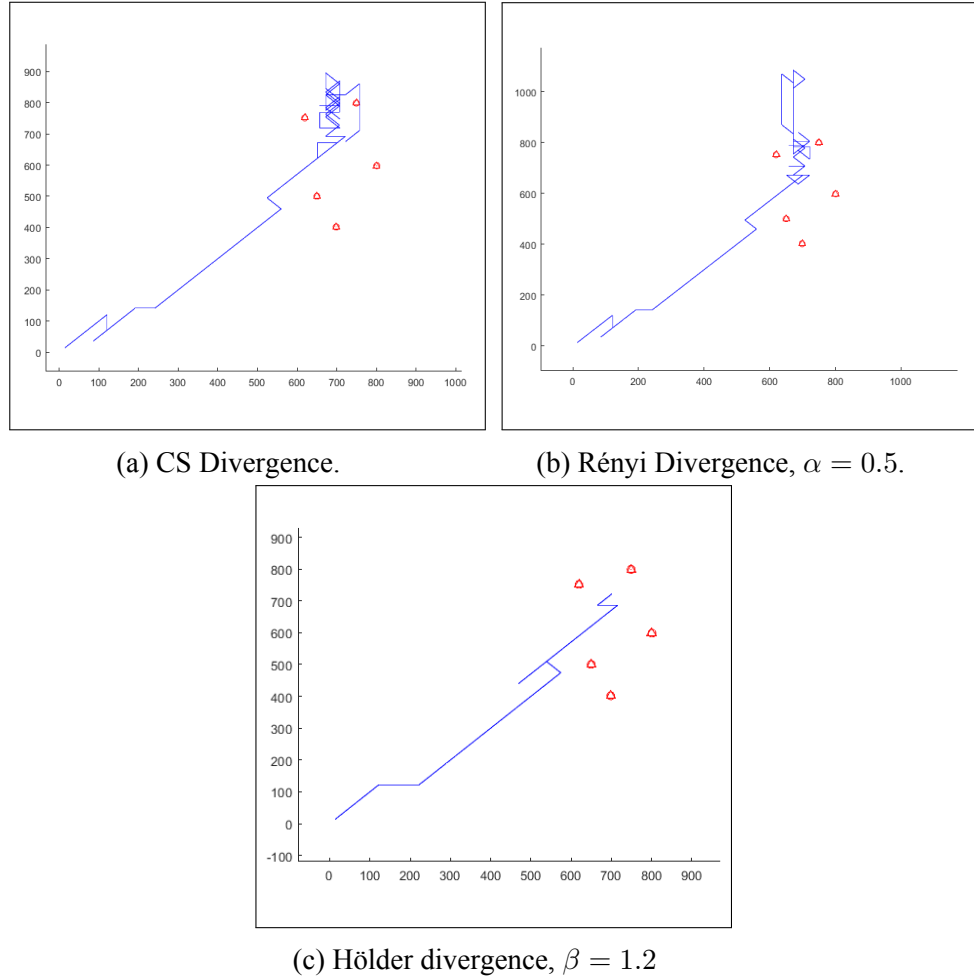
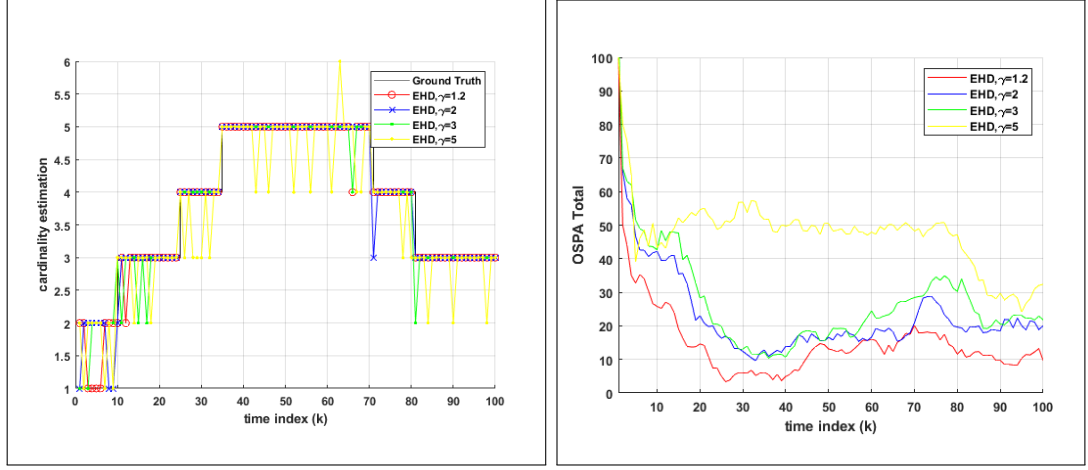


Figure 4.9: Sensor locations for different reward functions.

We further analyze the effect of the γ parameter value in the same scenario. In Hölder divergence, $\gamma > 0$ is a prescribed free scalar parameter defining a bi-parametric



(a) Cardinality estimation.

(b) Total OSPA error.

Figure 4.10: Effect of γ parameter selection on the sensor control performance, $\beta = 1.2$.

proper Hölder divergence, as seen in (4.19), which is adopted in this paper. As can be seen in (4.19), in the derived EHD expression, γ appears in the exponent of intensity functions of the predicted and posterior distributions. While α and β are related to each other, they can be freely used to emphasize the information gain between two distributions. The effect of the γ parameter on the performance of the sensor control is shown in Figure 4.10. From these results, it can be said that the range $1 < \gamma < 3$ would be a good choice in dynamic scenes.

4.4.1.3. Scenario 3

This is a more challenging scenario with 10 targets, entering and leaving the surveillance area at different time instants. The targets are still moving relatively close to each other thus creating a complex scene.

For the scenario shown in Figure 4.11, in which 10 targets move in different directions, and enter and leave the scene at different times. We compared the performance of the proposed EHD-based sensor control scheme against the other methods in terms of OSPA metric [96]. We also present the result of a stationary sensor case, as a baseline performance. Results are presented as the average of the 100 Monte Carlo runs. The point clouds shown in blue are the birth densities modeled as a Gaussian mixture with four components.

Figure 4.12 shows the ground truth and the average of cardinality estimates

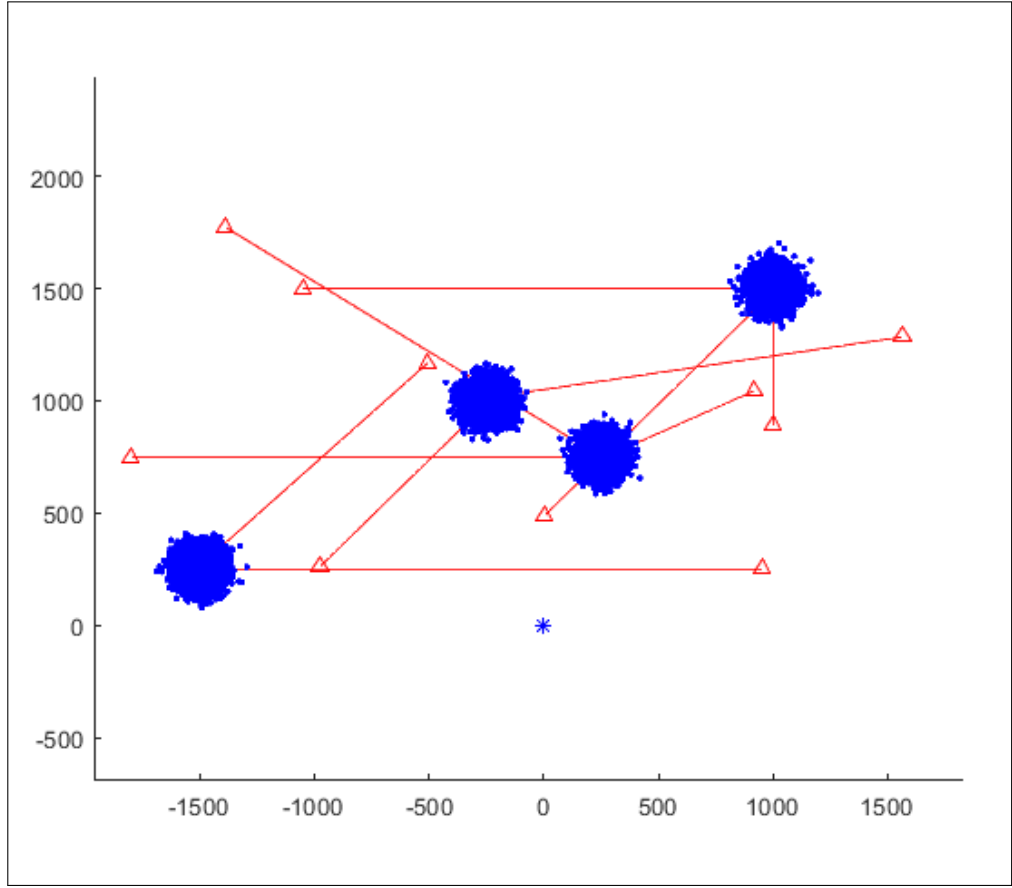


Figure 4.11: Scenario-2 ground truth trajectories. The targets are born inside the particle clouds and die at the red triangles.

evolving in time. The number of targets in the surveillance area is constantly changing and thus challenging the estimation filter to adapt to the current situation. It can be seen in Figure 4.12 that the performance of the proposed EHD-based reward function is superior to the other reward functions. It can be observed that it is adapting relatively fast to the changes in cardinality. While the performance of Rényi and CSD-based reward functions are particularly poor between time indices 40 and 100, where the number of targets increases from 5 to 10 and then decreases to 6, the performance proposed EHD is better in that period. Also, observe that the variations in the estimated cardinalities are smaller for the proposed reward function, compared to the other two methods. Therefore, in terms of cardinality estimation, it can be said that the proposed method is more robust to the changes in the number of targets in the surveillance area.

In Figure 4.13, the cardinality component of the OSPA metric is shown for the three reward functions. The superior performance of the proposed reward function can be seen in the figure, in terms of OSPA cardinality error. While the cardinality errors of the Rényi and CS divergence-based reward functions are high, especially in the second

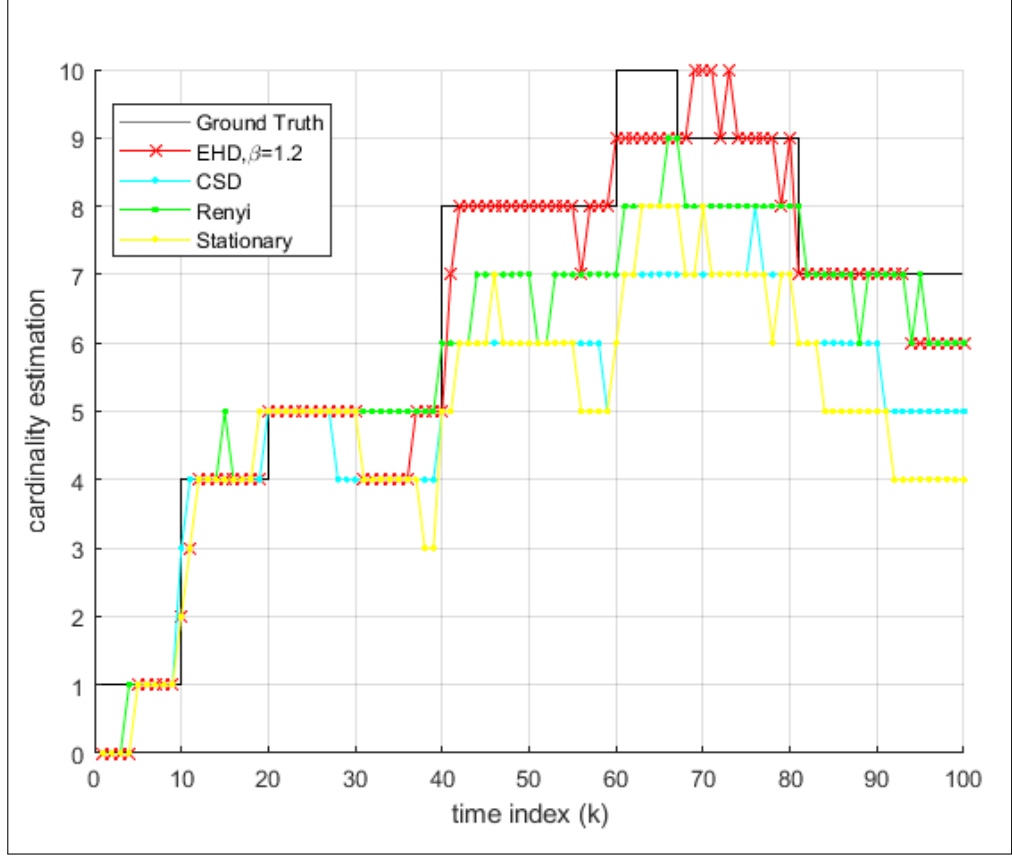


Figure 4.12: Cardinality estimation result of different information-theoretic reward functions.

half of the simulation, the proposed method achieves better performance.

This complex tracking scenario further shows that the proposed EHD is more effective than similar information-theoretic reward functions and its parameter selection can be used to provide flexibility to the control algorithm. Note that, we have selected the parameters as reported in [172] and have not done any parameter tuning. More significant results may be obtained by optimizing the parameters and creating a more suitable scenario. In this study, we present the preliminary observations on the proposed method.

Figure 4.14 shows the OSPA localization errors for the three reward functions. It can be seen that the proposed reward function has a smaller OSPA localization error than the other two.

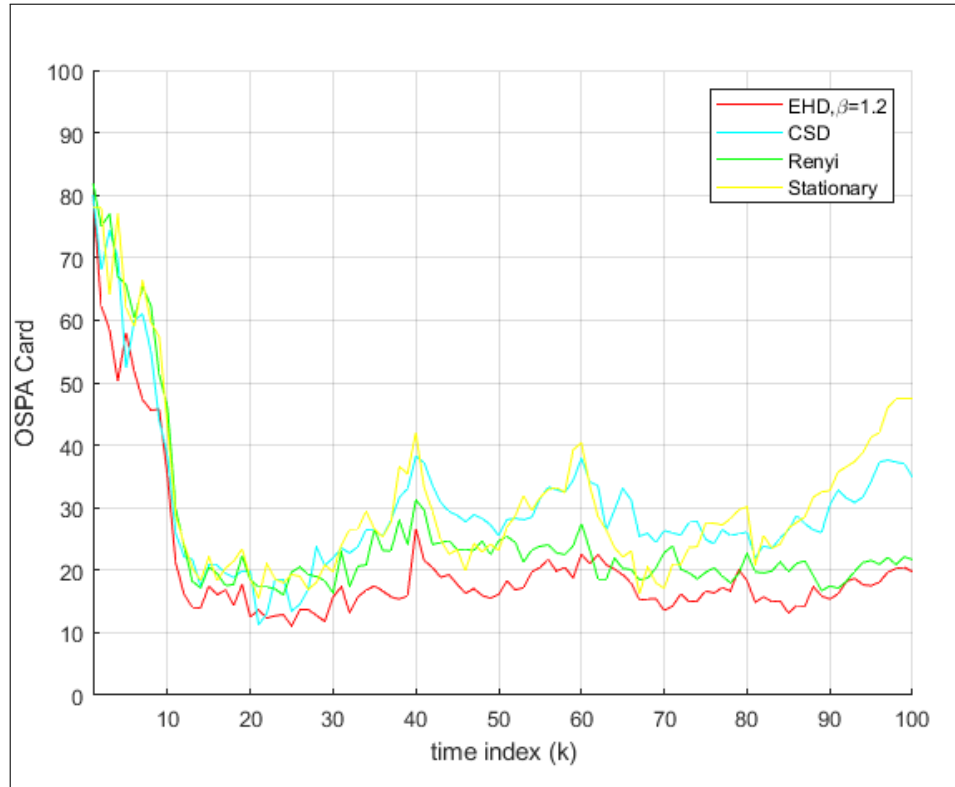


Figure 4.13: OSPA Cardinality Error Comparison.

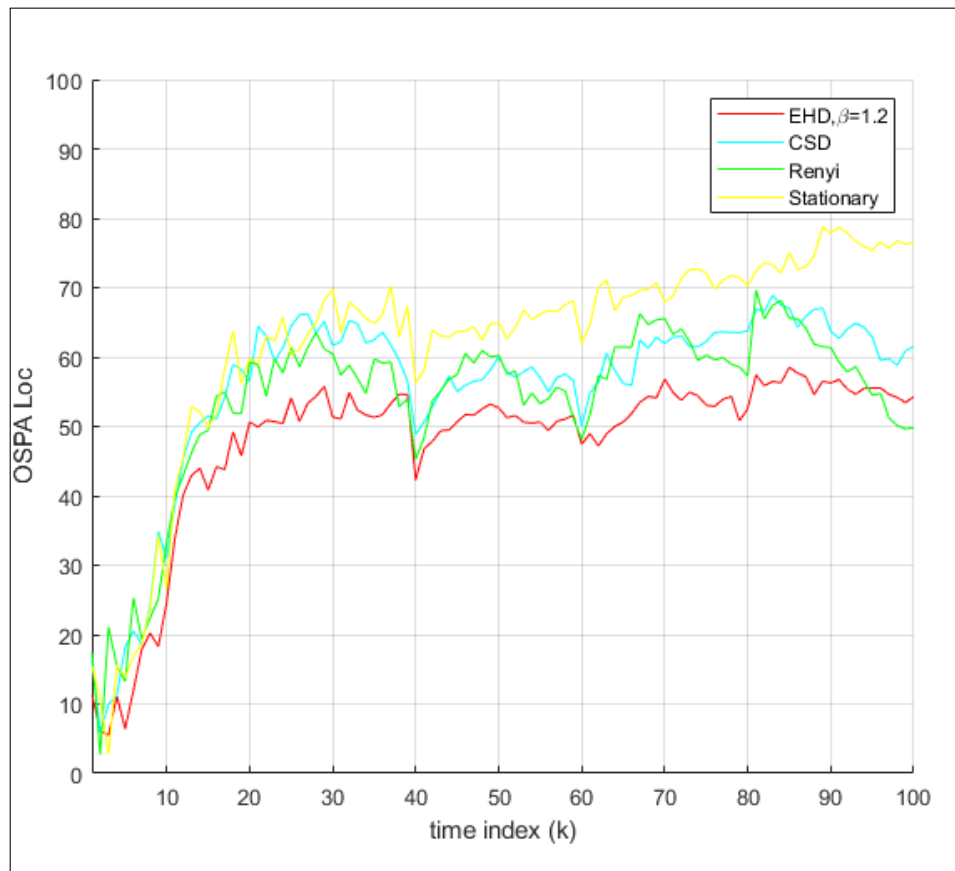


Figure 4.14: OSPA Localization Error Comparison for $\beta = 1.2$.

Table 4.1: Comparison of average OSPA errors for three reward functions.

Method	Average (in time) OSPA Error		
	Cardinality	Localization	Total
EHD, $\beta = 1.2$	20.5	47.8	68.3
CSD	30.3	55.7	86
Rényi, $\alpha = 0.5$	25.7	53.7	79.4
Stationary	31	61.9	84.7

4.5. Conclusions

This chapter presents a new information-theoretic reward function for sensor management in multi-target tracking based on the novel extended Hölder divergence using RFS theory. The Hölder divergence for the standard probability distributions is extended to the multi-object distributions. Specifically, the analytic expressions for the Poisson RFS case are derived. Implementations for both SMC and GM were presented and applied to a multi-target tracking problem using an SMC-PHD filter.

The performance of the proposed reward function is evaluated, in terms of the OSPA metric, and compared against widely used, similar reward functions in a challenging multi-target scenario. It is shown to be superior to the other methods. It is also shown that, by analyzing the obtained probability of the detection profile, the proposed reward function is better at finding an arguably optimum position for the sensor to detect all targets. Table 4.1 shows the time-averaged errors of each scheme. We have investigated the effect of parameters of EHD and Rényi based divergences on the sensor control performance and observed that the proposed reward function can be used to adapt the algorithm for different objectives and situations such as detecting targets or improving track quality, which can be hard for information-theoretic divergences in general because their physical interpretation not being clear in a tactical sense.

4.6. Appendices

4.6.1. Derivation of (4.31)-(4.33)

Recall that the Poisson RFSs have the form:

$$\begin{aligned}
p(X) &= e^{-\langle u, 1 \rangle} [Ku]^X \\
q(X) &= e^{-\langle v, 1 \rangle} [Kv]^X
\end{aligned} \tag{4.48}$$

Now, assume that the intensity functions $u(x)$ and $v(x)$ are modeled as Gaussian mixtures, such as:

$$u(x) = \sum_{i=1}^N \omega_i \mathcal{N}(x; \mu_i, \Sigma_i), \tag{4.49}$$

$$v(x) = \sum_{j=1}^M \omega_j \mathcal{N}(x; \mu_j, \Sigma_j). \tag{4.50}$$

Then the inner product $\langle u^\gamma, 1 \rangle$ of (4.19) is

$$\begin{aligned}
\langle u^\gamma, 1 \rangle &= \int [u(x)]^\gamma dx \\
&= \int \left[\sum_{i=1}^N \omega_i \mathcal{N}(x; \mu_i, \Sigma_i) \right]^\gamma dx
\end{aligned} \tag{4.51}$$

Using the result of Lemma 1, we obtain

$$\int \left[\sum_{i=1}^N \omega_i \mathcal{N}(x; \mu_i, \Sigma_i) \right]^\gamma dx \approx \int \sum_{i=1}^N \omega_i^\gamma c(\gamma, \Sigma_i) \mathcal{N}(x; \mu_i, \frac{\Sigma_i}{\gamma}) \tag{4.52}$$

$$= \sum_{i=1}^N \omega_i^\gamma c(\gamma, \Sigma_i) \underbrace{\int \mathcal{N}(x; \mu_i, \frac{\Sigma_i}{\gamma}) dx}_{=1} \tag{4.53}$$

$$\langle u^\gamma, 1 \rangle = \sum_{i=1}^N \omega_i^\gamma c(\gamma, \Sigma_i). \tag{4.54}$$

The same procedure is applied to the term $\langle v^{\gamma/\beta}, 1 \rangle$. The inner product $\langle u^{\gamma/\alpha}, v^{\gamma/\beta} \rangle$ of (4.19) is

$$\langle u^{\gamma/\alpha}, v^{\gamma/\beta} \rangle = \int \left[\sum_{i=1}^N \omega_i \mathcal{N}(x; \mu_i, \Sigma_i) \right]^{\frac{\gamma}{\alpha}} \left[\sum_{j=1}^M \omega_j \mathcal{N}(x; \mu_j, \Sigma_j) \right]^{\frac{\gamma}{\beta}} dx \tag{4.55}$$

$$\begin{aligned}
&\approx \int \left[\sum_{i=1}^N \omega_i^{\frac{\gamma}{\alpha}} c\left(\frac{\gamma}{\alpha}, \Sigma_i\right) \mathcal{N}\left(x; \mu_i, \frac{\Sigma_i}{\gamma/\alpha}\right) \right] \\
&\quad \cdot \left[\sum_{j=1}^M \omega_j^{\frac{\gamma}{\beta}} c\left(\frac{\gamma}{\beta}, \Sigma_j\right) \mathcal{N}\left(x; \mu_j, \frac{\Sigma_j}{\gamma/\beta}\right) \right] dx
\end{aligned} \tag{4.56}$$

If we bring together the summations and take them outside the integral in (4.56)

$$\begin{aligned} \langle u^{\gamma/\alpha}, v^{\gamma/\beta} \rangle &= \sum_{i=1}^N \sum_{j=1}^M \omega_i^{\frac{\gamma}{\alpha}} \omega_j^{\frac{\gamma}{\beta}} c\left(\frac{\gamma}{\alpha}, \Sigma_i\right) c\left(\frac{\gamma}{\beta}, \Sigma_j\right) \\ &\quad \cdot \int \mathcal{N}\left(x; \mu_i, \frac{\Sigma_i}{\gamma/\alpha}\right) \mathcal{N}\left(x; \mu_j, \frac{\Sigma_j}{\gamma/\beta}\right) dx \end{aligned} \quad (4.57)$$

Then, by rearranging the terms in (4.57), we obtain the expression

$$\langle u^{\gamma/\alpha}, v^{\gamma/\beta} \rangle = \sum_{i=1}^N \sum_{j=1}^M \omega_i^{\frac{\gamma}{\alpha}} \omega_j^{\frac{\gamma}{\beta}} c\left(\frac{\gamma}{\alpha}, \Sigma_i\right) c\left(\frac{\gamma}{\beta}, \Sigma_j\right) \mathcal{N}\left(\mu_i; \mu_j, \frac{\alpha \Sigma_i + \beta \Sigma_j}{\gamma}\right). \quad (4.58)$$

4.6.2. Derivation of (4.38)-(4.40)

When the predicted and posterior intensity functions $u_{k+1|k}(x)$ and $v_{k+1|k+1}(x)$ ³ are modeled as particle systems as given in (4.36) and (4.37), respectively, the inner product between $u(x)$ and $v(x)$ can be written as follows.

$$\langle u^{\frac{\gamma}{\alpha}}, v^{\frac{\gamma}{\beta}} \rangle = \int [u(x)]^{\frac{\gamma}{\alpha}} \cdot [v(x)]^{\frac{\gamma}{\beta}} dx \quad (4.59)$$

Inserting (4.36) and (4.37) into (4.59) results in⁴

$$\begin{aligned} &\int [u(x)]^{\frac{\gamma}{\alpha}} \cdot [v(x)]^{\frac{\gamma}{\beta}} dx \\ &\approx \int \left[\sum_{i=1}^N \omega_{k+1|k}^{(i)} \delta_{x_{k+1|k}^{(i)}}(x) \right]^{\frac{\gamma}{\alpha}} \cdot \left[\sum_{i=1}^N \omega_{k+1|k+1}^{(i)} \delta_{x_{k+1|k+1}^{(i)}}(x) \right]^{\frac{\gamma}{\beta}} dx \end{aligned} \quad (4.60)$$

If the sums of delta functions raised to a power, because of the different delta functions having different supports, the cross terms vanish in (4.60), i.e. $\delta_{x_{k+1|k}^{(i)}}(x) \cdot \delta_{x_{k+1|k+1}^{(i+1)}}(x) = 0$. Then, the integral can be approximated by the following expression.

$$\begin{aligned} &\int [u(x)]^{\frac{\gamma}{\alpha}} \cdot [v(x)]^{\frac{\gamma}{\beta}} dx \\ &\approx \int \left[\sum_{i=1}^N \left(\omega_{k+1|k}^{(i)} \right)^{\frac{\gamma}{\alpha}} \delta_{x_{k+1|k}^{(i)}}^{\frac{\gamma}{\alpha}}(x) \right] \cdot \left[\sum_{i=1}^N \left(\omega_{k+1|k+1}^{(i)} \right)^{\frac{\gamma}{\beta}} \delta_{x_{k+1|k+1}^{(i)}}^{\frac{\gamma}{\beta}}(x) \right] dx \end{aligned} \quad (4.61)$$

³For the sake of clarity, we dropped the subscripts, indicating the time indices of predicted and posterior intensity functions in the derivations.

⁴Note that in the particle filtering update $x_{k+1|k+1}^i \triangleq x_{k+1|k}^i$

This expression further simplifies when the two sums are multiplied. For the same reason, the deltas of different sums vanish and the approximation simplifies to the following expression:

$$\int [u(x)]^{\frac{\gamma}{\alpha}} \cdot [v(x)]^{\frac{\gamma}{\beta}} dx \approx \int \sum_i \left(\omega_{k+1|k}^{(i)} \right)^{\frac{\gamma}{\alpha}} \left(\omega_{k+1|k+1}^{(i)} \right)^{\frac{\gamma}{\beta}} \delta_{x_{k+1|k}}^{\frac{\gamma}{\alpha}}(x) \delta_{x_{k+1|k}}^{\frac{\gamma}{\beta}}(x) dx, \quad (4.62)$$

$$= \sum_i \left(\omega_{k+1|k}^{(i)} \right)^{\frac{\gamma}{\alpha}} \left(\omega_{k+1|k+1}^{(i)} \right)^{\frac{\gamma}{\beta}} \int \delta_{x_{k+1|k}}^{\gamma(\frac{1}{\alpha} + \frac{1}{\beta})}(x) dx. \quad (4.63)$$

Recall that the Hölder exponents α and β satisfy the relationship $\frac{1}{\alpha} + \frac{1}{\beta} = 1$ and we approximate the final expression [163],[174].

$$\langle u^{\frac{\gamma}{\alpha}}, v^{\frac{\gamma}{\beta}} \rangle = \sum_i \left(\omega_{k+1|k}^{(i)} \right)^{\frac{1}{\alpha}} \left(\omega_{k+1|k+1}^{(i)} \right)^{\frac{1}{\beta}}. \quad (4.64)$$

The second and third inner products $\langle u^\gamma, 1 \rangle$ and $\langle v^\gamma, 1 \rangle$ of (4.19) are straightforward to derive:

$$\langle u^\gamma, 1 \rangle \triangleq \int [u(x)]^\gamma dx \approx \int \left[\sum_{i=1}^N \omega_{k+1|k}^{(i)} \cdot \delta_{x_{k+1|k}}^{(i)}(x) \right]^\gamma dx. \quad (4.65)$$

Again, when the sum of delta functions is raised to a power, the cross terms vanish due to different supports and the approximation simplifies to

$$\int [u(x)]^\gamma dx \approx \int \sum_{i=1}^N \left(\omega_{k+1|k}^{(i)} \right)^\gamma \cdot \delta_{x_{k+1|k}}^\gamma(x) dx, \quad (4.66)$$

$$= \sum_{i=1}^N \left(\omega_{k+1|k}^{(i)} \right)^\gamma \int \delta_{x_{k+1|k}}^\gamma(x) dx, \quad (4.67)$$

$$\langle u^\gamma, 1 \rangle \approx \sum_{i=1}^N \left(\omega_{k+1|k}^{(i)} \right)^\gamma. \quad (4.68)$$

By applying the same procedure for $\langle v^\gamma, 1 \rangle$, we obtain

$$\langle v^\gamma, 1 \rangle = \sum_{i=1}^N \left(\omega_{k+1|k+1}^{(i)} \right)^\gamma. \quad (4.69)$$

5. LEVEL 5 FUSION: SENSOR FUSION FOR ASSURED NAVIGATION

Integrated navigation is another application to which sensor fusion is extensively applied. Due to the repeated use of integral operation to obtain position information from inertial measurements, the errors inherent in the inertial sensors are accumulated over time and cause the position estimation to diverge from the true position. As a result, despite being extremely accurate over short time intervals, there is a need to assist the inertial sensor data with other sources, such as GNSS/GPS systems. In this way, it is possible to ensure high accuracy and reliable navigation performance in prolonged missions.

However, since the satellite aiding depends on the availability of external satellite signals, there is a high probability of being exposed to, intentional or otherwise, various interference effects such as jamming, spoofing, etc. Especially in military applications, this jeopardizes the safety and success of missions. Hence, providing navigation aid by using onboard sensors and appropriately fusing their respective information is imperative for assured navigation systems.

Motivated by this, in this chapter, we present a multi-sensor fusion application for robust navigation where GNSS/GPS aiding is unavailable. By fusing information from two different aiding schemes, namely terrain, and visual aiding, in a decentralized/distributed manner, we demonstrate that it is possible to maintain navigation accuracy, independent of time, for long time periods. Due to the use of onboard sensors, the overall navigation solution is robust to external interference effects. Also, owing to the distributed fusion scheme, the overall system performance is robust to the loss of either alternative aiding systems.

The chapter is organized as follows. In section 5.1., a summary and background on integrated and aided navigation are presented. In 5.2. terrain and visual aiding concepts are discussed in detail. In 5.3., the fusion architecture is presented for terrain and visual aiding schemes. Section 5.4. presents the simulation results and section 5.5. concludes the chapter.

5.1. Aided Navigation

In an inertial navigation system, the platform estimates its kinematics (position, velocity, and attitude information) using a set of inertial sensors, namely accelerometers, and gyroscopes. The inertial sensors measure the linear and rotational acceleration of the platform. Then by integrating these measurements numerically, the position, velocity, and attitude of the vehicle are estimated. These inertial sensors provide accurate positioning capability in a short time duration. However, internal errors (both random and systematic) in the inertial sensors are accumulated due to the nature of the integral operation. Therefore, in a long time period, the position estimation will diverge from the true position dramatically [180]–[182].

It is crucial for a navigation system to support these inertial sensors by other means. The main aiding system used in navigation applications is satellite systems, i.e. GNSS. Unlike inertial sensors, the GNSS provides accurate position information in long time intervals. Thus, it complements inertial sensors for high-accuracy navigation applications. Therefore the fusion of GNSS and inertial information is a standard application for the navigation problem. However, for various reasons, such as the satellite line of sight not being available, tactical spoofing or jamming for military applications, or in space applications, satellite information may not be available or not be accurate to use as an aiding system. This brings up the need to use alternative aiding systems in navigation applications.

5.1.1. Literature Summary

High accuracy and reliable navigation requirements are becoming crucial, especially for unmanned platforms, aviation, military, and space applications. With autonomous/unmanned platforms becoming widespread, the cost, size, and performance of navigation systems have become more important. However, the current navigation sensors alone are not capable enough to satisfy all these requirements. Thus, a common method is to use two or more, ideally complementary, high-accuracy sensor modalities together for reliable navigation [183].

The main information source for navigation is the inertial sensors, due to their short-term accuracy, high frequency, and reliability [184] features. These sensors are accelerometers and gyroscopes configured along three Cartesian axes. However,

the main problem with these sensors is that the initialization errors and sensor bias accumulated in the final navigation solution in time. This in turn causes large errors and unreliable solutions in long time intervals. Therefore, these sensors require other information sources to eliminate these errors in the final navigation solution [117],[183],[185]. The information from the second source is usually fused with the inertial navigation solution by a Kalman-based estimation filter.

The inertial navigation systems (INS) use GPS/GNSS systems as the main aiding source. The satellite signals are acquired via a receiver and a position estimation from these signals is calculated. Then this position estimation is fused with the integrated navigation solution. However, the positioning from the satellite signals is not accurate enough for short time intervals and/or low-speed vehicles, because the small displacements can not be resolved. Thus, the inertial sensors and satellite systems complement each other's weaknesses naturally and are commonly used as integrated navigation systems [186],[187]. Due to the satellite line-of-sight, buildings, or urban canyons, the satellite signals can be lost partially or completely. Moreover, it is susceptible to external interference such as spoofing and jamming [184]. There are also cases where satellite signals are basically not available, like indoor applications and space missions.

When GNSS signals are not available, it is crucial to support aiding information to inertial sensors, using other onboard sensor modalities, to maintain accurate navigation capability. For this purpose, using imaging sensors and cameras is an attractive alternative due to their small size, rich information, low-cost characteristics [185],[188]–[190]. While surveying a known region, the position and attitude (called camera pose collectively) can be estimated absolutely [191]. Besides, by tracking visual features in image sequences, the camera motion can be estimated in image space [192],[193]. Regarding accuracy and data speed, the inertial sensors and imaging sensors are coming forward as complementary modalities and are subject to active research recently.

Regarding the available prior knowledge and the environment, visual navigation systems can be divided into three categories: (i) mapless systems, (ii) map-based systems, and (iii) map-building systems [188].

The map-based systems store the prior spatial information of the navigation environment. These maps, in general, contain the 3D model of the environment and objects. However, because of the requirement of prior knowledge of the learned navigation region, these systems become obsolete when this information is not

available.

The map-building systems are used in robot applications in general and do not need any prior or learned information about the environment. In this problem, the platform simultaneously maps the area and localizes itself relative to the surrounding region (called SLAM). This generated map is usually represented by a point cloud obtained from a lidar sensor or a camera. SLAM-based methods are becoming a more active research topic recently, due to the increased computational capabilities and deep learning and there is a plethora of studies in the literature [194]–[199]. SLAM is a non-linear problem by nature and requires non-linear filtering techniques. In SLAM, the state vector contains both the kinematic information about the platform and the map feature vectors. In addition to this, while navigating the platform continuously observes new and different features and augments the state vector with new information. This causes the computational complexity to increase. However, if some of the state variables in the state vector are not non-linear, then the state vector can be divided into linear and non-linear parts and processed separately, thus decreasing the computational complexity and increasing the performance. This method is called Rao-Blackwellization [200].

In mapless systems, navigation is done without any knowledge regarding the environment. The unmanned vehicle tries to observe distinctive features and navigates relative to these features. The most common approaches for this type of problem are feature tracking [201]–[203] and by the optical flow, [204], [205] methods. The optical flow methods Horn-Schunk [206] and Lucas-Kanade [207] are the most common methods. Horn-Schunk is called the global method, while the Lucas-Kanade is called the local. Recently, the fusion of IMU information and optical flow is applied to hovering and descent maneuver problems [207]. Besides, with the dense calculation of optical flow, all the moving objects can be detected and tracked [206]. In the feature tracking methods, usually, the stationary features such as corners, points, and lines are tracked in sequential images, and relative motions of these features in the images are calculated [208].

Inter-image motion parameters can be transformed into different forms to update the inertial navigation solution. For example, in [201] by using the prediction from the inertial solution, the relative motion estimates can be transformed into position pseudo-measurements. However this method assumes perfect knowledge of the attitude information. In [209], the relative pose between two images is used as an average speed to increase navigation accuracy. In [210], by using three-view geometry

a new navigation-aiding algorithm is proposed. In this method, the overlapping features in three images are fused with inertial information. This method improves navigation observability and estimation accuracy [201].

There are other approaches for visual-aided navigation (VAN). In [210], a stereo camera-based novel visual aiding algorithm is proposed. By visual odometry, motion estimation can be done in real-time for 30 fps. In [211], using range measurements in a SLAM concept, feature-based robot navigation is proposed. In [212] a method for using vertical lines from the azimuth is proposed. A relationship between the roll angle and vertical lines passing through the azimuth is established. A measurement model for roll angle is proposed. Through real-world experiments, they observed that the proposed method improves attitude and position accuracy. In [213], in the GNSS-denied environments, a hybrid VAN is proposed to improve UAV navigation accuracy. They used Gaussian process regression to model the accumulated errors or unobservable features while GNSS is available and used that model to reduce the errors when GNSS is lost. [214]–[217] various deep learning methods such as convolutional neural network and recurrent neural network are used in a SLAM concept using the camera information for navigation.

In [218] the finite set statistics (FISST) approach, which is especially popular for multi-target tracking applications, is proposed for visual navigation. The FISST approach eliminates the data association problem effectively. Random finite set (RFS) methods handle practical error sources such as sensor noise, false alarm, detection probability, and bias systematically.

An unsolved problem with vision-based navigation is that the camera adapts to changing conditions of a real flight, especially at low altitudes and for a short time. In addition, when using a monocular camera, the depth information is lost due to the projective transformation used in the camera model.

Another aiding method used in inertial navigation is called terrain-aided navigation (TAN). The TAN system uses altitude measurements provided by a radar altimeter onboard the vehicle to compare to digital terrain elevation data (DTED map). The radar altimeter measures the distance between the ground level and the aircraft [219]. But the fundamental problem with the TAN is the radar altimeter beamwidth problem. TAN assumes the relative distance from the ground level is measured directly below the vehicle (nadir distance). However, the radar altimeter measures the distance to the closest point in the beam footprint. At low altitudes, the beamwidth would be narrow, but at higher altitudes (800m or higher) the radar beamwidth would be

wider, thus the uncertainty in the radar measurement is increased, leading to a greater horizontal position error. To overcome this problem, the use of an interferometric radar altimeter is proposed [220].

To acquire precise position information using TAN, a non-linear estimation problem must be solved. By using local linearization, EKF-based solutions have been proposed. However, because of the highly non-linear characteristics of the terrain, the EKF algorithm can diverge due to linearization errors. The Bayesian estimation algorithms such as particle filter or point mass filter can prevent this divergence problem [221]–[225].

The performance of the TAN depends on the terrain characteristics under the flight profile. The flat or repetitive terrain profiles provide little to no information since the change in height is low, thus leading to increased position uncertainty. However, rough terrain is ideal for terrain aiding since it provides rich and high-quality altitude information.

5.2. Terrain and Visual Aiding Applications

In this section, terrain and visual aiding schemes for inertial navigation are presented in detail. First, a generic process model which is usually obtained from INS error dynamics is presented. Since both TAN and VAN applications are based on the Bayesian approach, the process model is common to both aiding schemes. Then, the measurement model for each problem is presented and discussed separately. Due to the difference in the measurement process of each problem, the respective measurements are quite different, and as we shall see in the following sections, are complementary to each other.

5.2.1. Definition of the Dynamical Model

First, we start by defining the state vector of the vehicle. The typical 15-element state vector of the aircraft contains the following variables

$$x_k = \left[L \quad \lambda \quad h \quad v_n \quad v_e \quad v_d \quad \theta_{3 \times 1} \quad b_{3 \times 1}^a \quad b_{3 \times 1}^g \right]^T \quad (5.1)$$

(L, λ, h) defines the position of the aircraft in latitude, longitude, and altitude, respectively. (v_n, v_e, v_d) represents the velocity of the aircraft body in the north, east, and down directions. $\theta = [\theta_r, \theta_p, \theta_y]$ represents the Euler angles as roll, pitch, and yaw. b^a and b^g are the 3-axis biases of the accelerometer and gyroscope, respectively.

In the inertial navigation context, the change in state variables in time is defined as error propagation. The general continuous-time (nonlinear) formula of the dynamic state propagation is defined as follows

$$\dot{x}(t) = f_t(x(t), u(t)) \quad (5.2)$$

Since the details on the evaluation of the process model in the form of the transition matrix were presented in Ch. 2.3., it is omitted here.

5.2.2. Visual Aided Navigation

For visual aided navigation, there are two approaches:

- Using a database of known 3D geodetic landmark points and associating them to the detected 2D features on the image.
- Using associated 2D features between two consecutive images.

The first method results in an absolute pose estimation of the camera/platform, however, requires some knowledge of the navigation region in the form of a database. This database contains the geodetic coordinates of distinct features/landmarks in the region. Then, when a new image has been captured, some feature extraction method is applied and the resulting features are associated with the points in the database. By using these associated 3D-2D point pairs, the camera projection matrix and absolute camera pose can be estimated. The performance of the pose estimation depends on the accuracy of the association of the points.

The second method, also known as visual odometry, results in a pose estimation relative to a previous one, using associated 2D-2D feature pairs between successive camera frames. Therefore, the performance depends also on the accuracy of the previous pose estimation, in addition to the feature association accuracy.

We will present a detailed definition of the problem and the methods in the remainder of the chapter.

5.2.2.1. Problem Definition

Standard cameras use the so-called pinhole camera model to project a 3D world into the image space. In this model, when projecting the 3D world points onto the image plane, a ray is assumed to exist which is passing through both the projection center (the camera center) and the 3D point. This projection is modeled mathematically as follows:

$$\tilde{\mathbf{x}} = \zeta \mathbf{x} = P\tilde{\mathbf{X}} \quad (5.3)$$

In the above equation, the ζ is called the scale factor and represents the distance of the 3D point $\tilde{\mathbf{X}}$ to the focal plane of the camera. $P_{3 \times 4}$ is called the camera matrix and constructed as follows

$$P = K[R|t] \quad (5.4)$$

Where K , R , and t are called the camera calibration matrix, camera rotation, and camera translation, respectively.

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, t = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

When K is known, the other two parameters can be extracted from the camera projection matrix P . When the (5.3) is written in component form

$$\begin{aligned} u = \frac{\tilde{u}}{\zeta} &= \frac{p_{11}\tilde{X} + p_{12}\tilde{Y} + p_{13}\tilde{Z} + p_{14}}{p_{31}\tilde{X} + p_{32}\tilde{Y} + p_{33}\tilde{Z} + 1} \\ v = \frac{\tilde{v}}{\zeta} &= \frac{p_{21}\tilde{X} + p_{22}\tilde{Y} + p_{23}\tilde{Z} + p_{24}}{p_{31}\tilde{X} + p_{32}\tilde{Y} + p_{33}\tilde{Z} + 1} \end{aligned} \quad (5.5)$$

As can be seen from the above equation, to estimate the parameters of the matrix P , we need to find 11 parameters first. We have two independent equations and 11 free parameters. Thus, when at least 6 point pairs are known, the elements of the matrix P can be estimated. For this purpose, the following relationship is established

$$\begin{bmatrix} 0^T & -\zeta_i \mathbf{X}_i^T & y_i \mathbf{X}_i^T \\ \zeta_i \mathbf{X}_i^T & 0^T & x_i \mathbf{X}_i^T \end{bmatrix} \begin{bmatrix} \mathbf{p}^1 \\ \mathbf{p}^2 \\ \mathbf{p}^3 \end{bmatrix} = 0 \quad (5.6)$$

where $\mathbf{p}^1, \mathbf{p}^2, \mathbf{p}^3$ are rows of the projection matrix P in the column form. When n point pairs are known, (5.6) can be written as follows

$$\mathbf{A}\mathbf{p} = 0 \quad (5.7)$$

Where A represents $2n \times 12$ data matrix and $\mathbf{p}$ is the vectorized projection matrix with 12 elements. This is a least squares problem and when the point pairs contain noise it can be solved via non-linear least squares algorithms (NLS), specifically by the Levenberg-Marquardt NLS algorithm.

5.2.2.2. Camera Measurement Model

In this section, we will present the camera measurement model used in the vision-aided navigation problem. The general mathematical model of the camera measurements related to the state vector is as follows

$$\begin{aligned} z_i &= \begin{bmatrix} u_i \\ v_i \end{bmatrix} + \eta_i = h_i(x, \eta_i) = h_i(x) + \eta_i \\ x &= \begin{bmatrix} \delta P_I^G & \delta V_I^G & \delta \theta & \delta g_b & \delta a_b \end{bmatrix}^T \end{aligned} \quad (5.8)$$

with the following definitions:

- η_i represents the zero mean white Gaussian noise process.
- The elements of the state vector x are position, velocity, attitude errors of the INS, and gyro and acceleration biases of the IMU, respectively.
- u_i, v_i are the pixel location of the $i - th$ feature in the image plane.

The errors and the biases in the state vector are all 3×1 vectors with (x, y, z) components. Therefore, the state vector is a 15-element vector. $h_i(x)$ is, in general, a non-linear function of the state vector.

Next, we need to define the frames of reference used in the problem. Fig. 5.1 shows the relationship between the camera frame, navigation frame, and global frame. The geodetic landmark positions, $P(f_i)^G$, in the database, are in the global frame. We assume the camera and the IMU are aligned. The detected feature in the image, $P(f_i)^C$ is in the camera frame which is the image measurements. The vehicle/camera position in the global frame is given as P_I^G .

The first term in (5.11) corresponds to the change in the measurement w.r.t. the change in the camera position and is calculated as

$$\frac{\partial z_i}{\partial p_i^C} = \frac{1}{z_i'^2} \begin{bmatrix} 1 & 0 & -x' \\ 0 & 1 & -y \end{bmatrix} \quad (5.12)$$

The change in the camera position w.r.t. the change in the state vector is as follows:

$$\begin{aligned} \frac{\partial p_i^C}{\partial x} &= \begin{bmatrix} \frac{\partial p_i^C}{\partial C_G^I} & \frac{\partial p_i^C}{\partial P_I^G} \end{bmatrix} \\ &= \begin{bmatrix} KC_G^I [P_{f_i}^G - P_I^G]_{\times} \delta \theta - KC_G^I \delta P_I^G \end{bmatrix} \end{aligned} \quad (5.13)$$

Thus, combining (5.12) and (5.13) we obtain the final Jacobian of the pinhole measurement model as follows.

$$\frac{\partial z_i}{\partial p_i^C} = \frac{1}{z_i'^2} \begin{bmatrix} 1 & 0 & -x' \\ 0 & 1 & -y' \end{bmatrix} \begin{bmatrix} -K\hat{C}_G^I & 0_{3 \times 3} & K\hat{C}_G^I [P_{f_i}^G - P_I^G]_{\times} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \quad (5.14)$$

The zero blocks in the Jacobian matrix represent the assumption that pixel positions are not changing w.r.t. the velocity errors and gyro and acceleration biases.

We will use this model to estimate the camera position and attitude from the given 3D-2D point correspondences. To obtain this, the following NLS algorithm based on the Levenberg-Marquardt method is used.

5.2.2.3. Visual Aiding Application to Inertial Navigation

The previous section explained how the relationship between the INS error state and the camera measurements is established and how the position estimation of the camera is obtained. In this section, we will present the navigation-aiding algorithm based on the Kalman filter concept to reduce IMU errors and improve the accuracy of the navigation result. The algorithm block diagram is shown in Fig. 5.2. Note that, in the aiding algorithm, the state vector is defined with the error terms, not the absolute variables. Thus, the estimated errors in the result of the Kalman filter are fed back to the INS solution and then reset. The algorithm details are as follows. At each time instance, the IMU measures the acceleration and rotation rate of the vehicle. By integrating

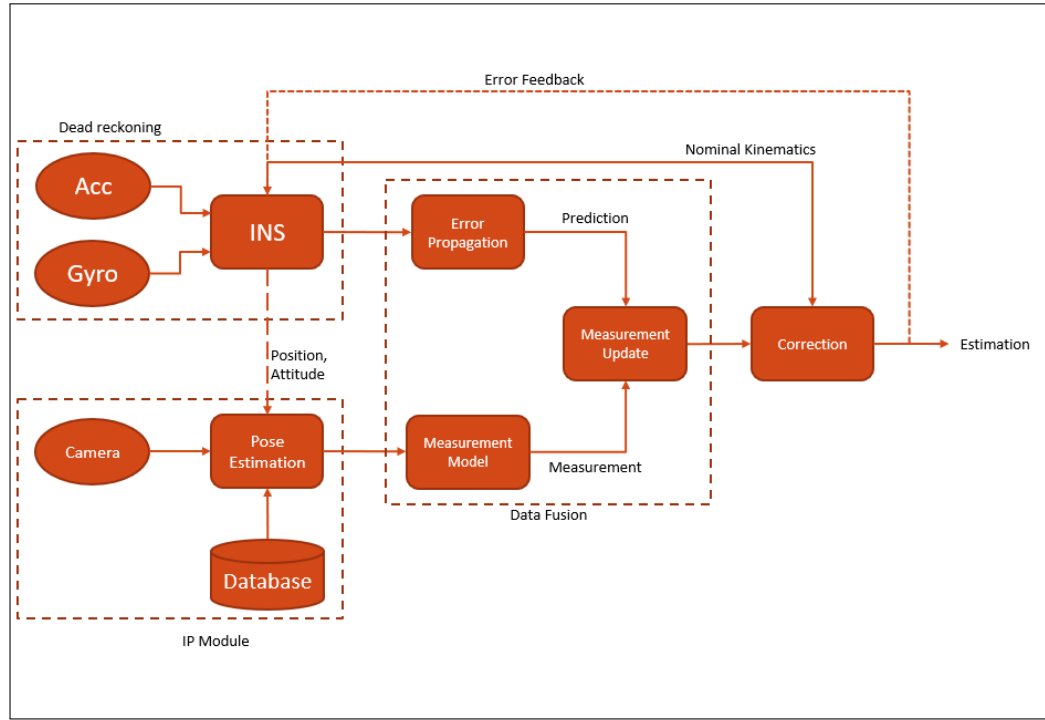


Figure 5.2: VAN algorithm block diagram.

these inertial measurements in time, an inertial navigation solution is produced by the strapdown INS. The solution includes the position, velocity, and attitude of the vehicle. The INS works at a high frequency (usually 100Hz). Due to the systematic characteristics of the IMU, the estimates of the INS are erroneous as well and these errors accumulate and grow boundless over time.

The physical process model of the Kalman filter is derived from the INS error propagation equations. This is a very standard operation and for the 15-state problem, the derivation and calculation of the state transition matrix F and its covariance matrix Q can be found in [37] and also were given in Ch. 2.3. At the time update instance, the state transition matrix, and its covariance is calculated and the current error-state vector and the estimation error covariance is propagated in time.

At a lower frequency than the IMU, the camera captures an image, usually looking down at the Earth's surface. On the captured image, some distinct features are detected and matched to the landmark points in the database, and from these correspondences, the camera position is estimated to be used as a measurement in the Kalman filter update. Using the Kalman filter measurement update procedure, the best estimates of the kinematics and IMU errors and their covariance are calculated. Thus, a navigation fix is obtained.

Obtained errors are subtracted from the INS output and the corrected navigation

states are obtained. After the estimated errors are used to fix the INS output, the Kalman state is reset to zero vector. This is because the current errors are assumed to be estimated and after the fix, the cycle begins from the start.

5.2.3. Terrain Aided Navigation

Terrain aiding is another alternative to the GNSS for high-accuracy navigation applications. In this section, we will present the mathematical definition of the TAN problem and algorithm design.

5.2.3.1. Problem Definition

In TAN, a combination of a radar altimeter and a digital elevation map (DEM/DTED) are used to calculate a position fix to use in aiding the inertial navigation system. The radar altimeter measures the relative distance between the surface and the vehicle. Ideally, this distance is assumed to be the nadir distance of the vehicle. The difference between the altitude of the vehicle and the terrain height information from the DTED map results in the height estimate from the ground level. And the difference between this and the radar altimeter measurement gives the final height error estimate.

The radar altimeter measurement at the measurement time k is modeled as follows

$$z_k = h_k^{INS} - h_k^{RA} = h(L, \lambda) - \tilde{h}_k + \eta_k \quad (5.15)$$

where η is the white noise, z_k is the measured height from the terrain point, RA represents the radar altimeter, $h(\cdot)$ is the height of the terrain at the horizontal position indicated by (L, λ) , latitude and longitude respectively, which is obtained from INS. $\tilde{h}$ is the altitude error of the INS output. Fig. 5.3 shows the radar altimeter measurement concept in TAN. The relationship between the measurement and state variable, i.e. position of the vehicle, in (5.15) is nonlinear, due to the DTED map function $h(\cdot)$. Thus, either a linearized model of the corresponding function or a non-linear filter must be used to estimate the state of the vehicle.

While, ideally, it is assumed that the radar altimeter measures the relative distance as nadir distance, in practice, the radar altimeter measurement is the distance

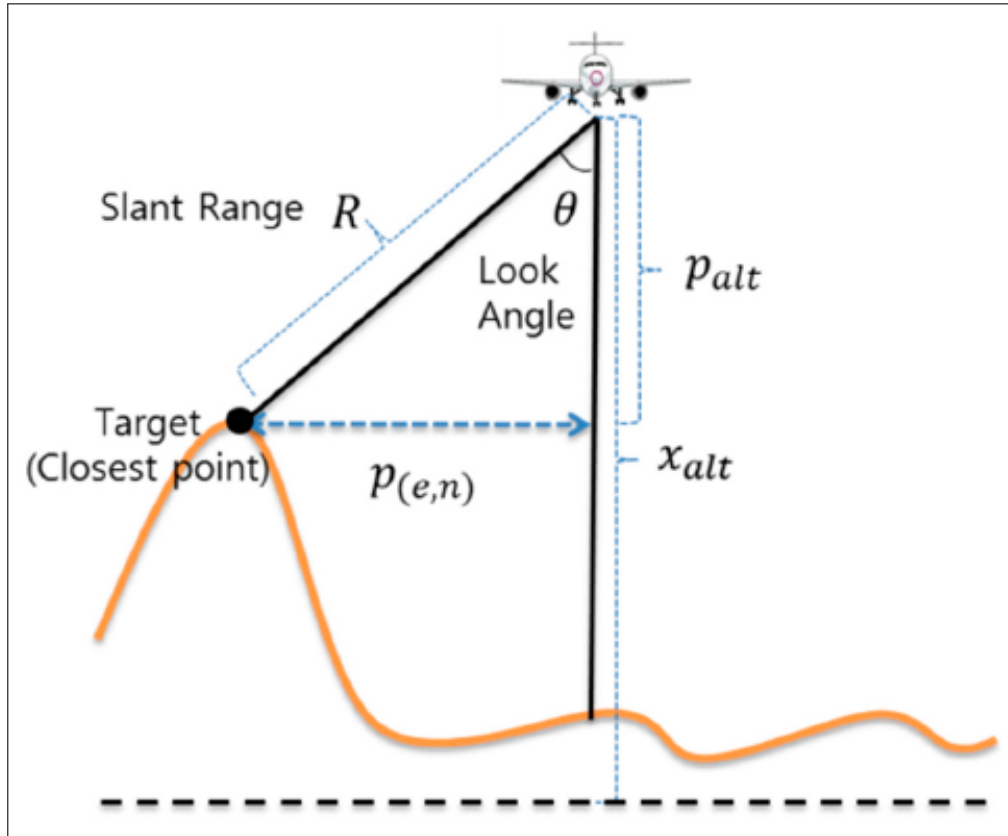


Figure 5.3: Radar altimeter measurement in TAN.

of the closest point on the surface to the vehicle as seen in Fig. 5.3. In the figure, the beamwidth of the radar altimeter is denoted as θ . While, ideally, the RA should measure the height x_{alt} from the point directly below the aircraft, it actually measures the height p_{alt} from the closest point. Thus, there would be an offset in the lateral position on which the aircraft thinks it is. This offset introduces a measurement error in the estimation process.

Due to the beamwidth of the radar signal and the vehicle altitude, the radar illuminates an area on the surface. When there are objects like trees or higher than the ground-level terrain structures in the illuminated area, the returned signal is from the closest point of the surface to the vehicle. This causes an extra horizontal localization error, especially at high altitudes, because the assumption is that the return signal would be from directly under the vehicle.

Using this radar altimeter measurement and the nominal position information, we can estimate the terrain elevation at the point (L, λ) . Then, using the DTED map, we can obtain the digital elevation value at the same point. By comparing those values, we can estimate the position error and use it to fix the navigation output using a Kalman-based (or a Bayesian estimator) filter. Since the measurement model relating

the state and the radar altimeter measurement (the DTED map) is a non-linear function of horizontal position, we must use a linearized filter or preferably a non-linear filter to estimate the errors with high accuracy.

5.2.3.2. DTED Maps

DTED map is a 2D matrix that for every latitude/longitude stores the mean sea level (MSL) height of the surface of the Earth. Between the $\pm 70^\circ$ latitudes, has the same resolution in both latitude and longitude. DTED maps are classified according to their resolutions as follows:

- DTED Level 0: 10 Arcsec resolution (900 meters)
- DTED Level 1: 3 Arcsec resolution (90 meters)
- DTED Level 2: 1 Arcsec resolution (30 meters)

Every DTED map covers $1^\circ \times 1^\circ$ area between the latitudes $\pm 70^\circ$. Therefore, a DTED map contains 1200×1200 elevation values.

The most reliable open-source DTED database is the database created by NASA during the Shuttle Radar Topography Mission (SRTM) mission. This database contains Level 1 and 2 DTED maps of the whole world. SRTM maps are generated using interferometric radar on the satellites.

5.2.3.3. Radar Altimeter Measurement Model

The measurement equation relating the aircraft position and the radar altimeter reading in (5.15) is non-linear in general. This non-linearity stems from the terrain height being non-linear w.r.t. distance. This measurement model needs to be linearized in order to be used in the Kalman filter framework (i.e. EKF).

The linearization techniques for the given model are known as terrain stochastic linearization [226],[227]. Consider the terrain profile function $h(\cdot)$ which is expanded to the Taylor series, at the predicted position, i.e. INS position.

$$h(L, \lambda) = h(\hat{L}, \hat{\lambda}) + \frac{\partial h}{\partial L}(L - \hat{L}) + \frac{\partial h}{\partial \lambda}(\lambda - \hat{\lambda}) \quad (5.16)$$

where the partial derivatives represent the terrain slope in the northward ($\frac{\partial h}{\partial L}$) and eastward ($\frac{\partial h}{\partial \lambda}$) directions. The shorthand notations for the partial derivatives are h_L and h_λ .

The linearized plane equation is given as

$$h_{lin}(L, \lambda) = n_L(L - \hat{L}) + n_\lambda(\lambda - \hat{\lambda}) + T. \quad (5.17)$$

where n_L and n_λ are the surface normal components. These parameters can be computed as

$$\begin{aligned} n_L &= \frac{\partial h}{\partial L} \big|_{(\hat{L}, \hat{\lambda})} \\ n_\lambda &= \frac{\partial h}{\partial \lambda} \big|_{(\hat{L}, \hat{\lambda})} \\ T &= h(\hat{L}, \hat{\lambda}) \end{aligned}$$

Then, minimizing the quadratic error between (5.16) and (5.17) over a grid patch, the parameters of the linearized plane equation n_L , n_λ , and T can be optimized:

$$\arg \min_{n_L, n_\lambda, T} \sum_i (h(L_i, \lambda_i) - (n_L(L_i - \hat{L}) + n_\lambda(\lambda_i - \hat{\lambda}) + T))^2 \quad (5.18)$$

In the system equation form, when combined with (5.15), the linearized (error form) measurement equation can be written as

$$\delta z = \begin{bmatrix} n_L & n_\lambda & -1 \end{bmatrix} \begin{bmatrix} \delta L \\ \delta \lambda \\ \delta h \end{bmatrix} + \eta \quad (5.19)$$

5.3. Multi-Sensor Fusion for Assured Navigation

In this section, we focus on the state-level fusion of inertial-visual-terrain data for a high-accuracy navigation application, especially when the GNSS information is not available. As described in previous sections, the inertial measurements can be fused with visual and terrain information separately to obtain a stable navigation solution. However, the two schemes have their individual advantages and shortcomings.

Therefore, we will investigate fusing the results of these aiding systems to obtain a more robust and accurate navigation solution. The aim is to maintain a stable, accurate navigation solution, when GNSS signals are not available, hence the term "assured navigation".

Both aiding methods use the inertial solution in their time prediction step. Thus, they have a common process model. Due to the common process noise model and the feedback mechanism, the estimated state of each sensor modality would be correlated [228]. This correlation would be handled by the geometric average (GA) or arithmetic average (AA) fusion algorithms. Fig. 5.4 shows the proposed architecture for the navigation fusion application.

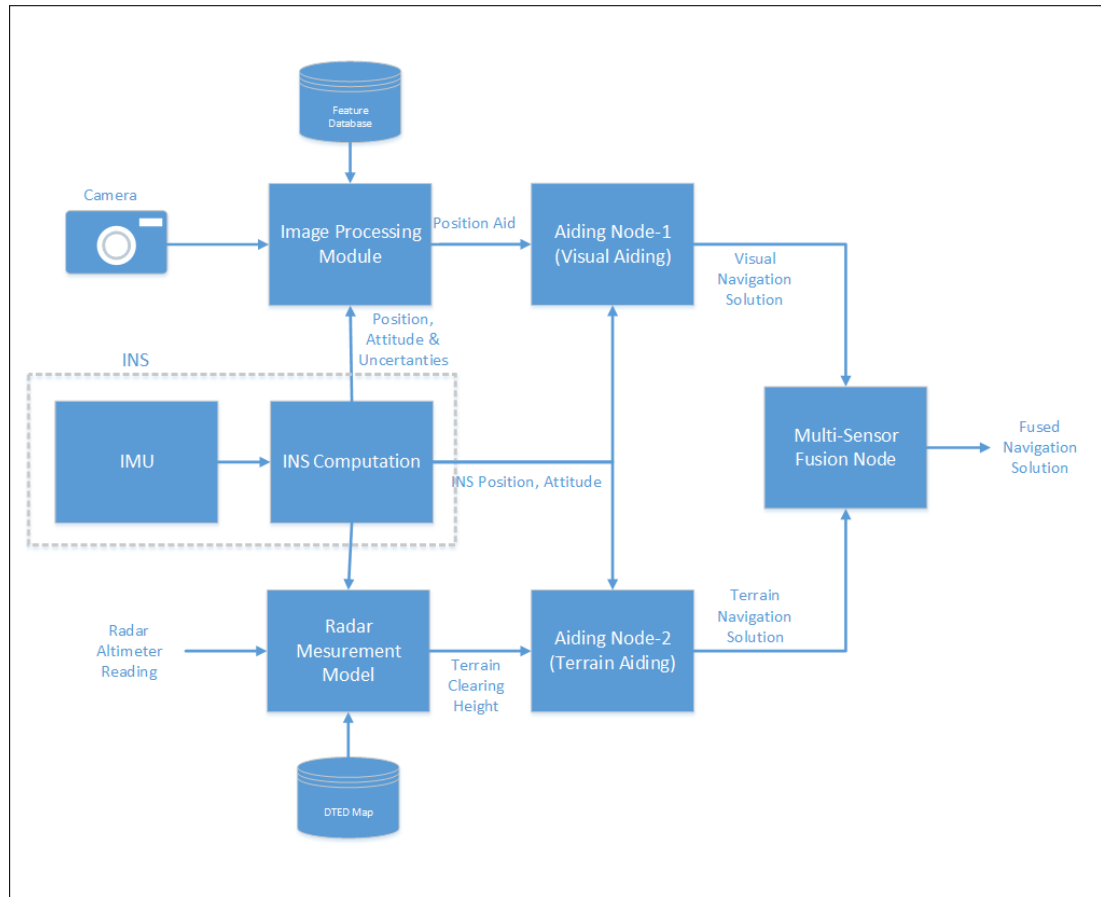


Figure 5.4: Multi-sensor distributed fusion architecture for assured navigation.

We apply the GA to obtain a robust navigation solution where GNSS aiding is not applicable. The proposed architecture is robust to data incest problems and the availability of individual sensor data due to environmental effects. It is also scalable regarding the addition or removal of individual sensor suites. By simulations, we evaluate the performance of the fusion result and compare it against the individual

sensor data. Also, we show that the proposed distributed architecture maintains navigation accuracy even when one of the sensor modalities is not available.

The next section presents the numerical results.

5.4. Numerical Studies

The distributed fusion architecture described in the previous section is implemented and evaluated with different scenario parameters and trajectories. There are many parameters that affect the outcome of any simulation run. However, only the critical parameters are described here.

The first parameter is flight height. The measurements from both visual and terrain aiding are directly affected by the height. We have generated trajectories at $2500m$ high. In visual aiding, the height has an impact on the perspective transformation of the camera model. As the height from the ground increases, the area covered by each pixel increases, thus the impact of a one-pixel error in the final positioning accuracy increases. Similar reasoning can be established for terrain aiding. Within the operating limits of the radar altimeter, increasing the height increases the ground coverage and the measurement error model in turn.

We have collected all the error sources of the inertial sensors into the "IMU quality" since the individual inertial sensor error sources such as scale factor, bias, stability, misalignment, etc are out of our scope. The quality of IMU is another parameter that directly affects navigation accuracy. For higher-grade IMUs (navigation grade), the errors are much smaller w.r.t. to the lower-grade IMUs (tactical grade). Therefore, while for lower grade IMUs the navigation accuracy decreases faster between the measurement updates, for higher grade IMUs accuracy decreases slower.

The flight duration affects the navigation accuracy as the accuracy of the free-inertial solution increases without bound with time. Thus, it is important to show that long-term navigation accuracy is bounded as a result of external aiding. Therefore, we expect the estimated position errors to be bounded.

The weights of the fusion method are another important parameter as it basically decides how much the fusion algorithm trusts the estimation of individual sensor modalities.

Other than the aforementioned parameters, there are parameters that are kept constant during simulations. These parameters are presented in Table 5.1. The

Table 5.1: Sensor fusion scenario parameters.

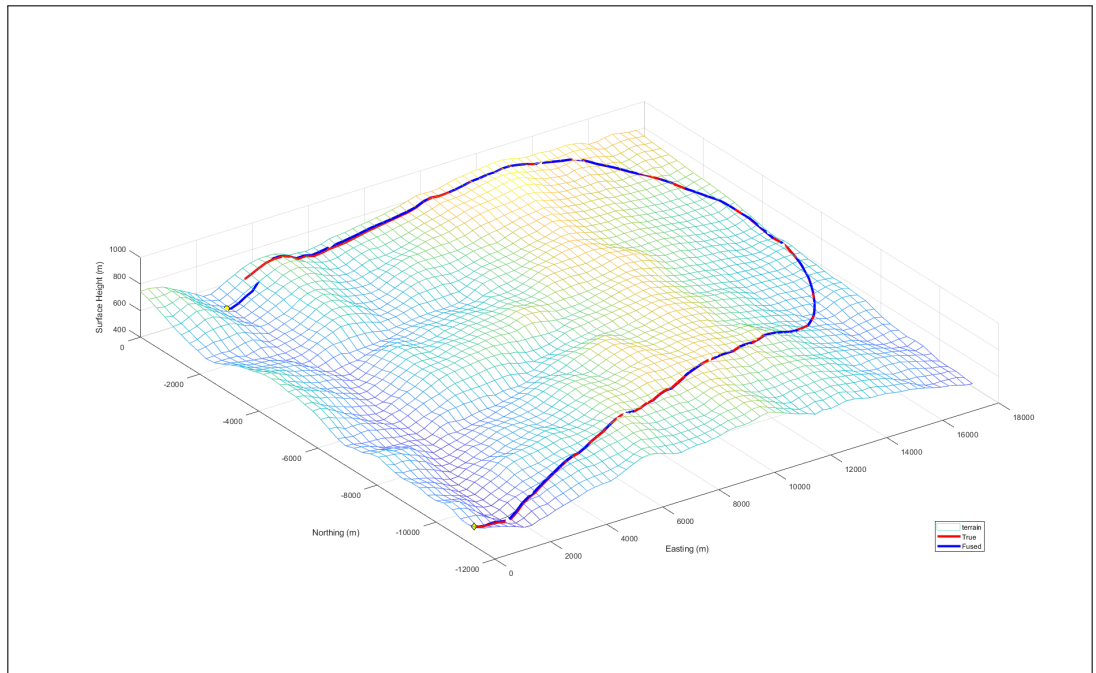
Parameter	Value	Parameter	Value
x_{init}	$[39.9, 32.3]^\circ$	σ_R	$100m$
L_{limits}	$[38, 41]^\circ$	σ_I	$1pixel$
λ_{limits}	$[28, 31]^\circ$	$N_{feature}$	10
v_{body}	$[200, 0, 0]m/s$	$z_{feature}$	$50m$
ϕ_{att}	$[0, 0, 90]^\circ$	f_{IMU}	$100Hz$
IMU	tac/nav	$\Delta t_{prediction}$	$0.1s$
ω_{GA}	0.5	Δt_{update}	$5s$

descriptions of the parameters given in Table 5.1 are as follows. x_{init} is the initial position of the aircraft in the horizontal plane, given in latitude and longitude, respectively. L_{limits} and λ_{limits} define the flight in geodetic coordinates as latitude and longitude, respectively. v_{body} is the velocity vector of the aircraft in body frame coordinates, and ϕ_{att} represents the attitude of the body frame. IMU grade is either tactical or navigation in simulations. ω_{GA} is the fixed fusion weight and is kept constant if not stated explicitly. σ_R and σ_I are the standard deviations of the radar altimeter and image feature measurements, respectively. $N_{feature}$ is the number of features used in camera pose estimation and $z_{feature}$ is the height interval inside of which the detected image features are assumed to be spread. $\Delta t_{prediction}$ and Δt_{update} are the prediction and update periods used in the estimation process. These represent the typical scenario parameters and their values used in simulations. In general, there are other parameters such as terrain characteristics and profile, or flight trajectory that affect the navigation-aiding performance but are hard to parameterize. Those parameters are described in the simulation scenarios when necessary.

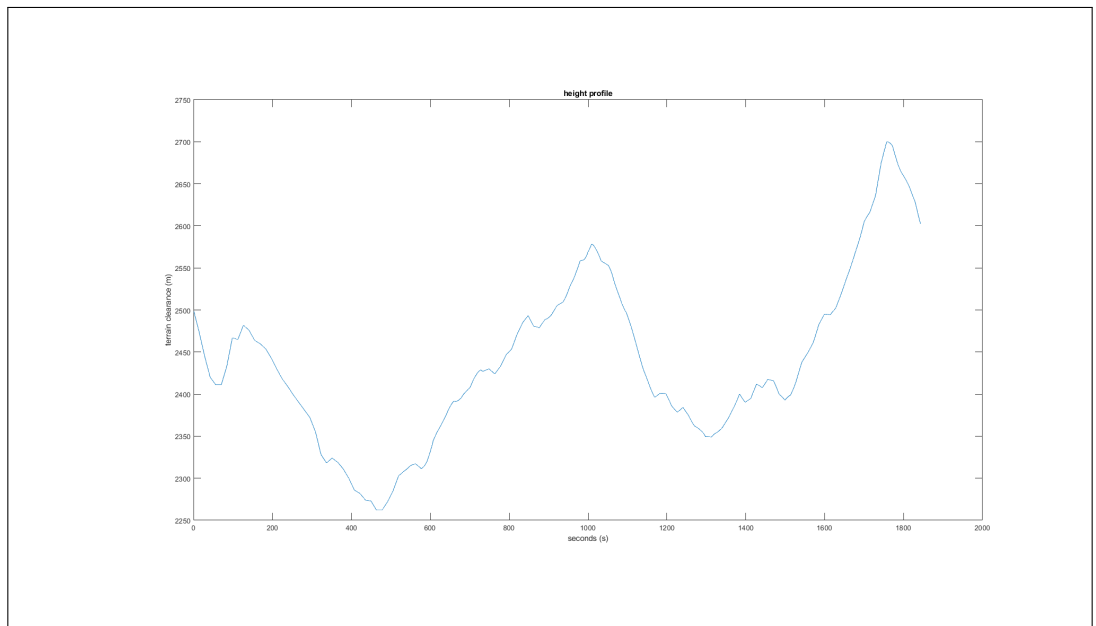
5.4.1. Scenario-1

In this scenario, a typical, relatively short-duration flight trajectory is simulated with the described parameters. The inertial measurements are assumed to be generated by a tactical grade IMU at $100Hz$. The ground-truth flight trajectory and the estimated trajectory are shown in Fig. 5.5a and the flight altitude profile 5.5b. The flight starts from the yellow circle and makes a U maneuver and ends at the yellow diamond point. Flight duration is approximately $180s$.

In Fig. 5.6, error bounds (σ bounds) and actual errors in each channel are shown.



(a) Trajectory side-view.



(b) Flight profile.

Figure 5.5: Scenario-1 ground truth (red) and estimated trajectory (blue).

As expected, the errors are inside the error bounds. Note that, this is the error of a single run. Smoother results can be obtained via Monte Carlo simulations.

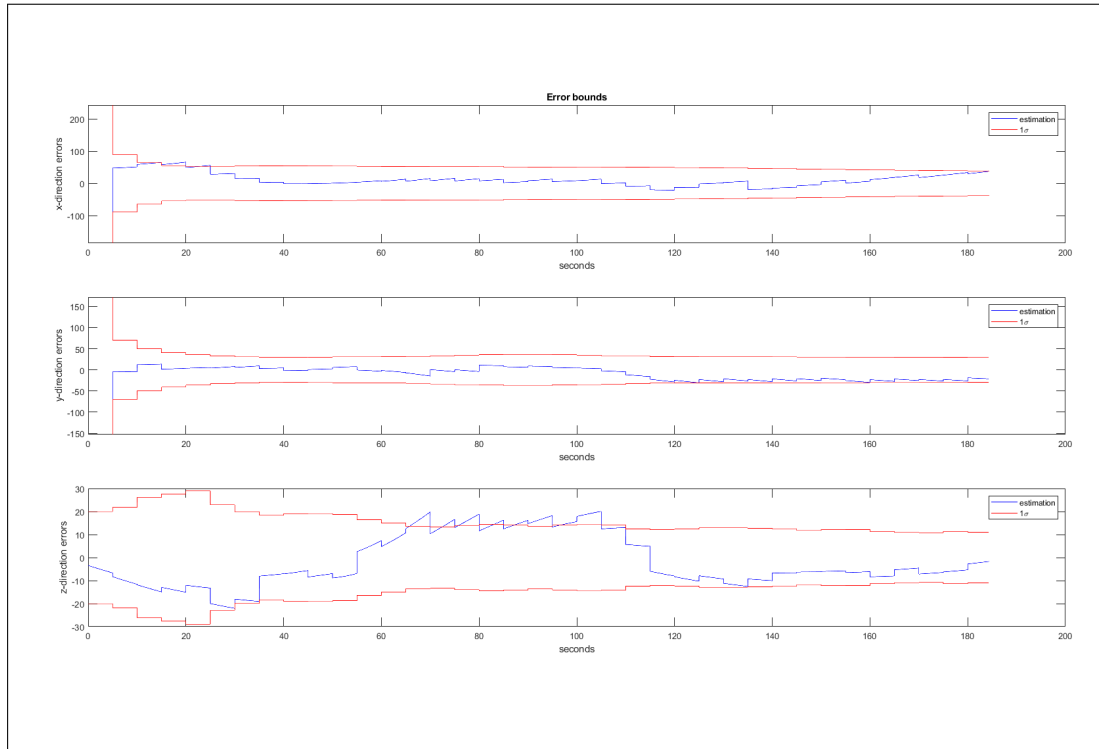


Figure 5.6: Scenario-1 error bounds and actual error.

The position components of the covariance matrices of VAN, TAN, and fused solutions are shown in Fig. 5.7. The saw-tooth effect in the plots is caused by prediction and update intervals being different. For this reason, the positioning error increases with consecutive time updates and when a measurement (position fix) is received and a measurement update is applied, the error decreases.

In the figure, the first two rows show the 3σ bounds on northing, easting, and altitude errors for VAN and TAN solution, respectively. The third row shows the error bounds of the GA fusion method. As can be seen from the figure, the effectiveness of the fusion is obvious. Because of the terrain characteristics, the TAN solution is less accurate than the VAN solution in the horizontal plane. However, since in the TAN aiding scheme, the height from the ground is directly measured, the altitude channel is very accurate. In contrast, for VAN, while the horizontal channel is very accurate, due to a lack of depth information in image features, the altitude estimation is less accurate. In the fused estimation, even when using the non-optimal fusion weight, the resulting solution is better than individual solutions. It is possible, even when using a tactical

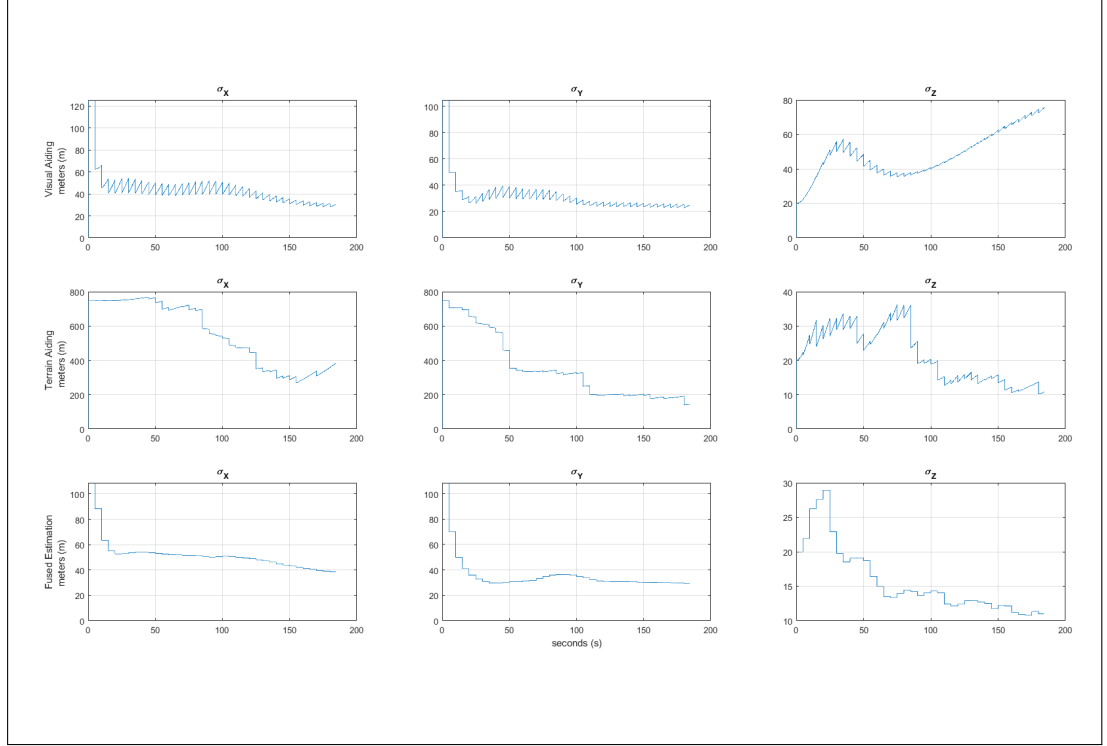


Figure 5.7: VAN (top row), TAN (middle row), and GA fusion (bottom row) 3σ bounds.

grade IMU, to obtain a horizontal accuracy of less than $50m$ and vertical accuracy of less than $30m$.

The distributed fusion architecture provides reliable navigation output even when one of the sensing suites is unavailable, at any time, for any reason. This can be seen in the limit cases when $\omega_{GA} \rightarrow 0$ and $\omega_{GA} \rightarrow 1$, effectively choosing the result of either solution. Besides, the GA fusion method bypasses the need to evaluate the time correlation between the individual sensor estimations and prevents double counting information due to the common process noise (or model) used in both sensor modalities.

At every $0.1s$, Kalman time update is applied on the error state vector and covariance matrix. At every $5s$, it is assumed, for each aiding system, a measurement is obtained and measurement update equations are applied. Note that the saw-tooth effect on the covariance components. It is caused by the difference between prediction and update periods. Until a new measurement is obtained and a measurement update is applied, the errors caused by integrated inertial measurements are accumulated. When a new measurement is observed and a state update is done, estimated errors (inertial biases) are subtracted from the current bias. Thus, reducing the estimation uncertainty instantly.

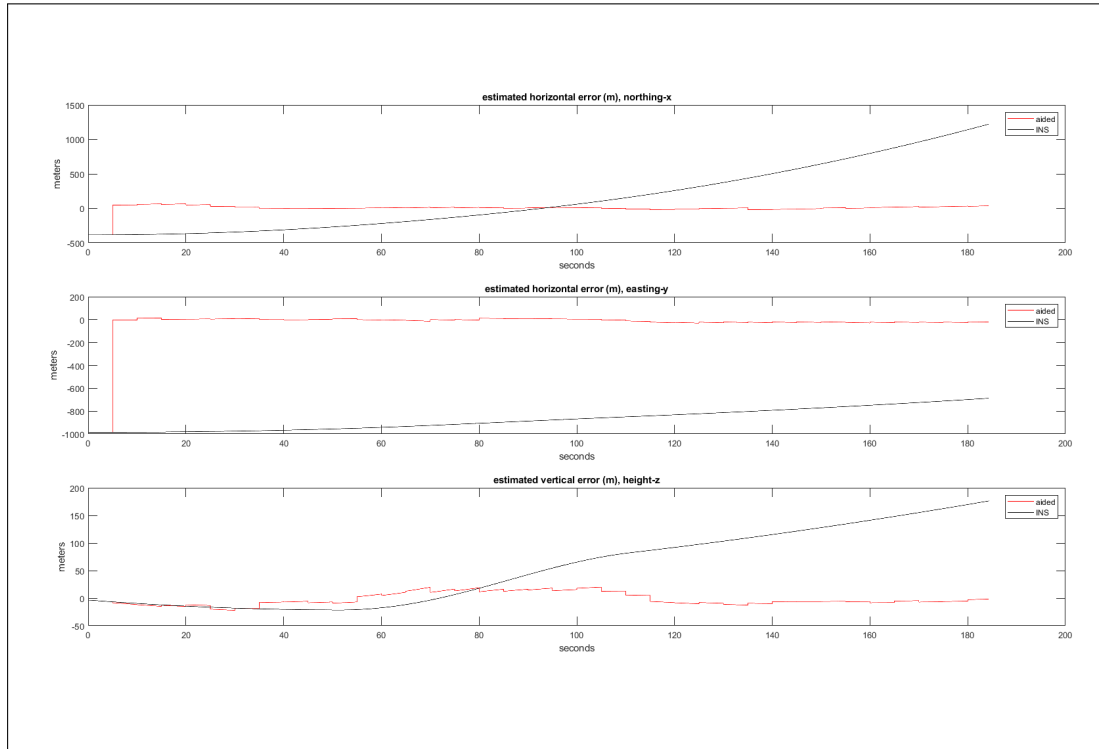


Figure 5.8: Scenario-1 free inertial error and fused estimation error.

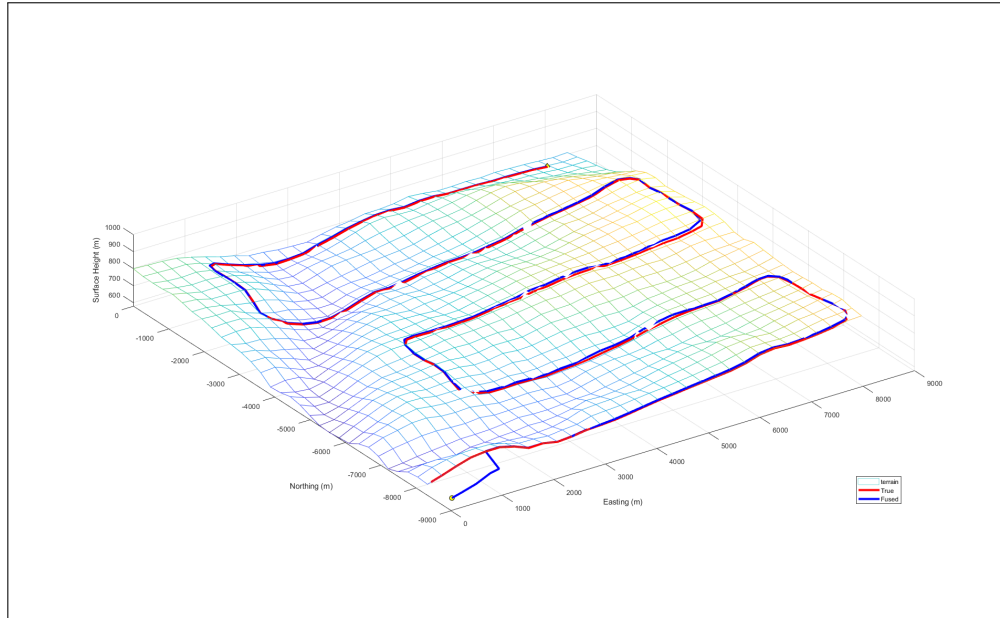
The fused estimation vs free inertial solution is shown in Fig. 5.8. The error in free inertial solution is, as expected, increasing without bound. The effectiveness of the multi-sensor fusion is easily seen in the figure.

5.4.2. Scenario-2

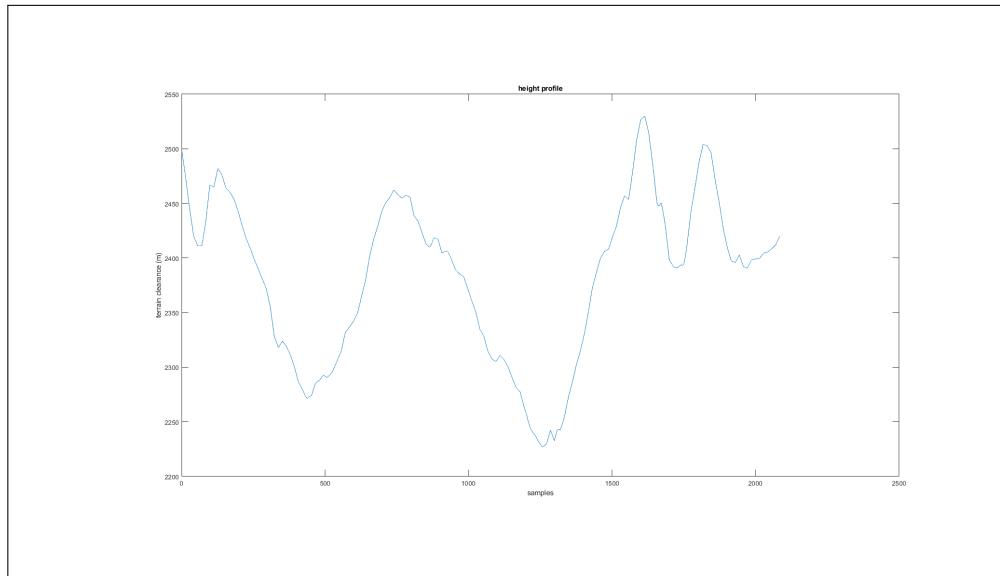
In the second scenario, the same simulation parameters are used, however, the trajectory consists of several maneuvers and flight legs. The trajectory along with the estimation result is shown in Fig. 5.9a. In the figure, the blue color represents low-altitude terrain characteristics, while the yellow color represents higher-altitude regions. The flight starts at the yellow circle and travels towards the yellow diamond point. Note the difference between the blue and red lines at the beginning of the scenario. Until the first measurement update (and fusion update), initial errors are present. After that, the position error is reduced dramatically.

In Fig. 5.10, the error bounds obtained in scenario 2 are shown for each channel as a result of the fusion process. Again, actual errors, even for a single run, are inside the bounds.

In Fig. 5.11, the position components of the estimated covariance matrix for



(a) Trajectory side-view.



(b) Flight profile.

Figure 5.9: Scenario-2 ground truth (red) and estimated trajectory (blue).

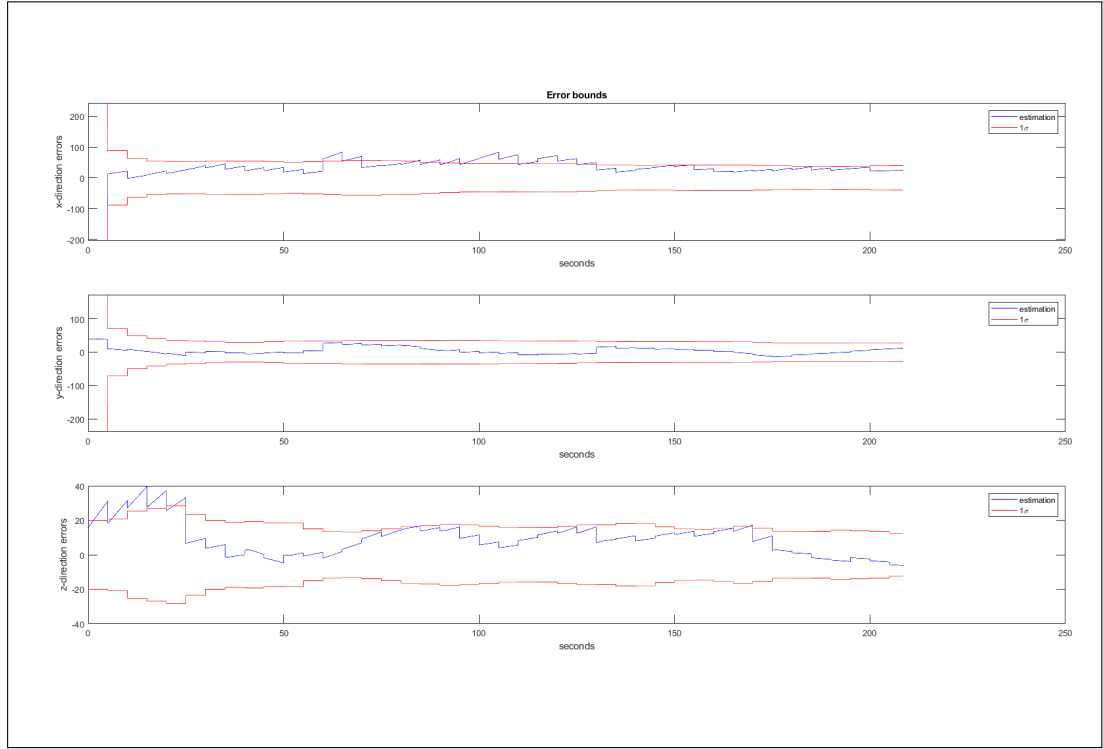


Figure 5.10: Scenario-2 error bounds.

VAN, TAN, and the fused solution are shown. The X , Y , Z axes are corresponding to northing, easting, and altitude, respectively. VAN solution, as can be seen from the figure, performs quite well on the horizontal channel, providing $< 50m$ accuracy. However, because the feature spread assumption is low w.r.t. the flight altitude, the height covariance is increasing. On the other hand, the height estimation of the TAN method is again very accurate, but because of the terrain characteristics, the lateral estimation accuracy is lower than the VAN. In the fused estimate, we have obtained a solution that is better than the individual aiding systems, approximately lower than $50m$ in all three channels.

In Fig. 5.12, plots for the free INS solution and fused estimation errors are shown for Scenario 2. Note that the horizontal errors in the free INS are increasing. The altitude error, on the other hand, seems to be oscillating. This is because, without any aiding mechanism, the altitude channel is unstable. Therefore, usually, the altitude channel is stabilized using a barometric altimeter, even in the free inertial case.

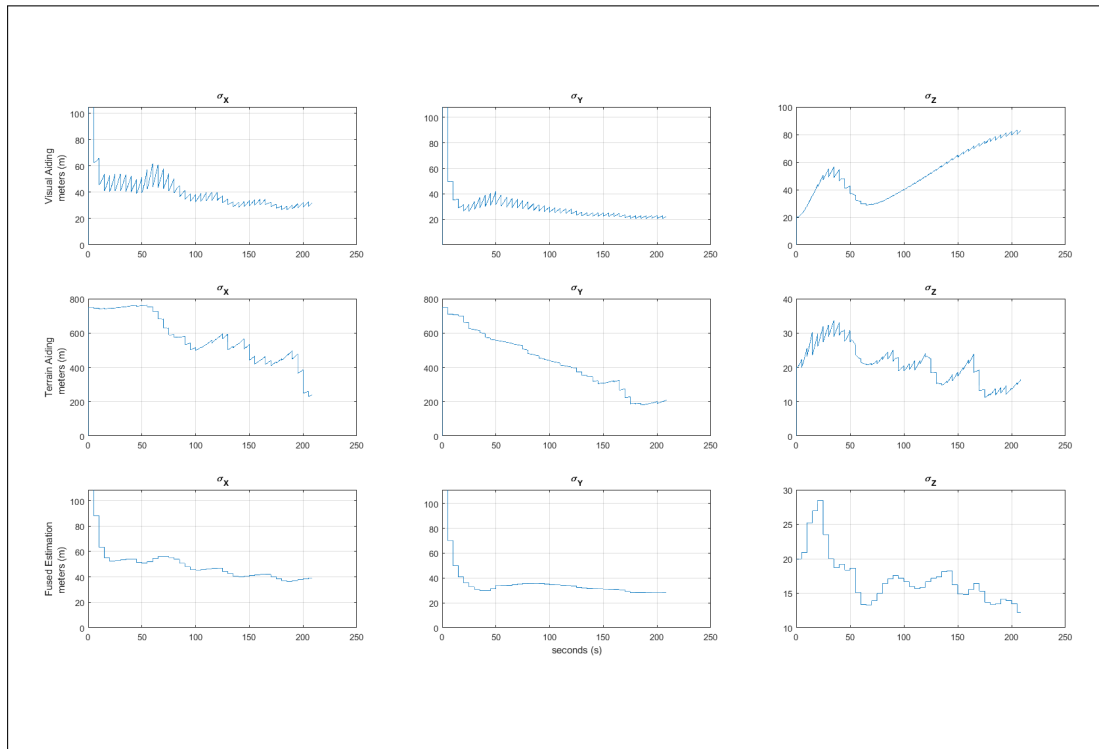


Figure 5.11: VAN (top row), TAN (middle row), and GA fusion (bottom row) 3σ bounds.

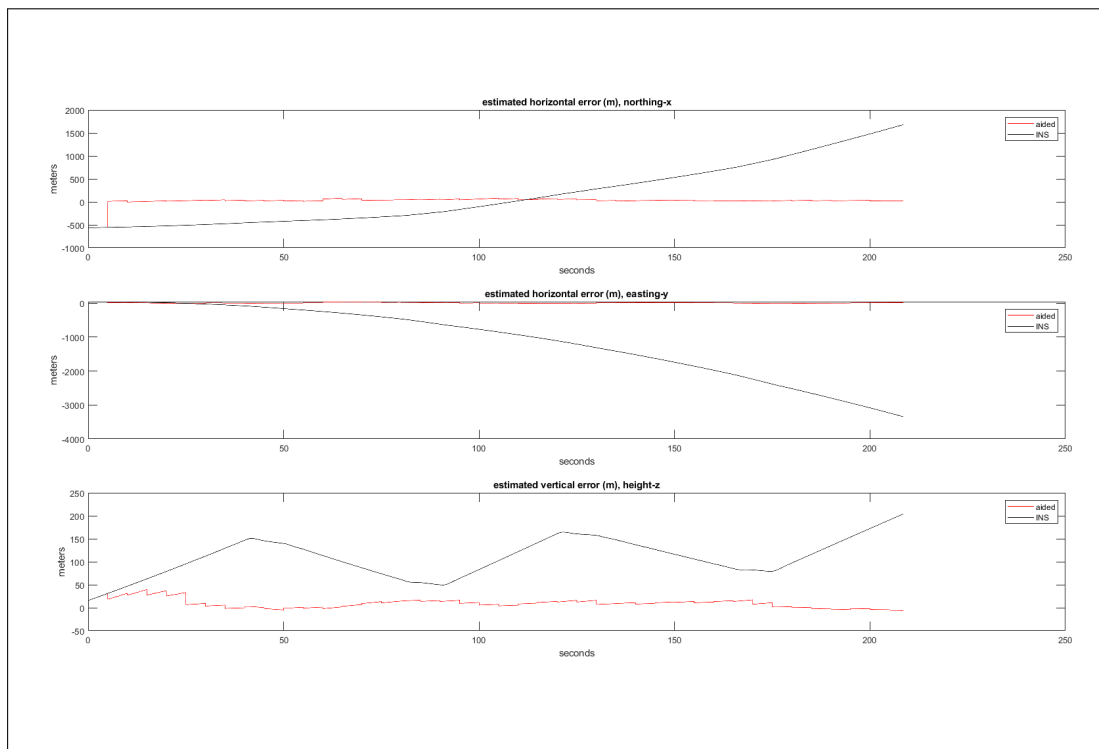


Figure 5.12: Scenario-2 free inertial error and fused estimation error.

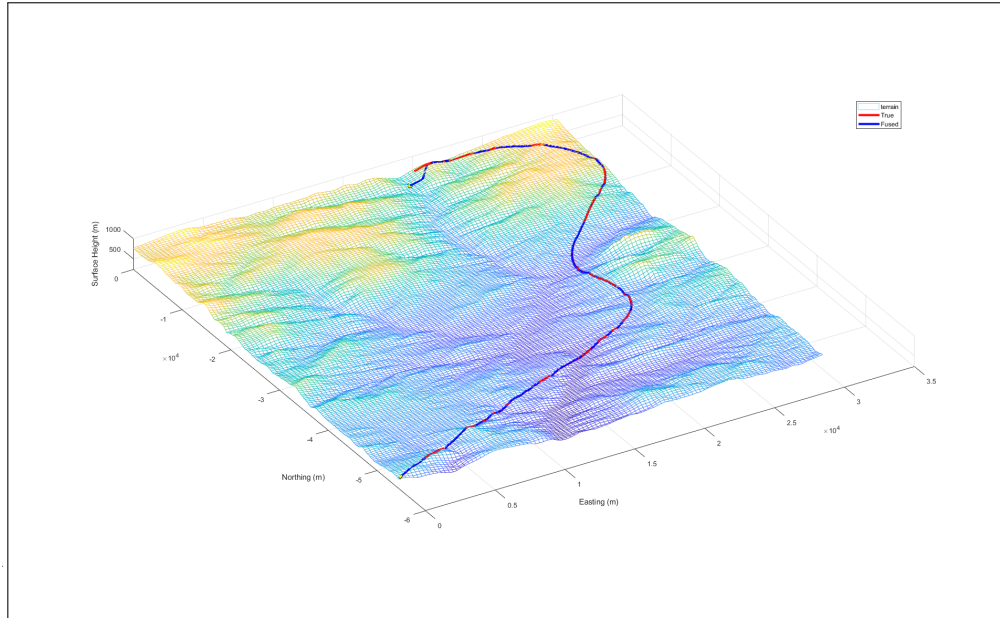
5.4.3. Scenario-3

Scenario 3 presents a longer flight and a different maneuver. The flight trajectory is shown in Fig. 5.13a and the flight profile under the trajectory is shown in Fig. 5.13b. Again, the flight starts at the yellow circle (upper edge of the region) and travels towards the diamond point lowermost of the region. The change in height in the flight profile is suitable for TAN to better localize the platform in the first part. Thus, we see an accuracy increase during that time (see Fig. 5.15 TAN result).

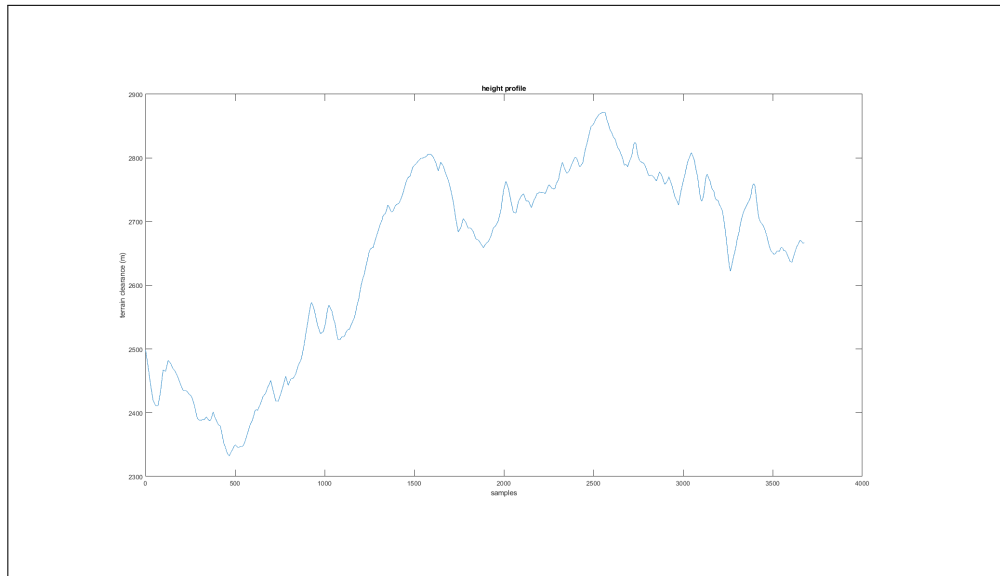
The estimation error bounds for scenario-3 are given in Fig. 5.14. We can see the estimation accuracy is $< 20m$ vertical and approximately $50m$ horizontal.

In Fig. 5.15 the individual estimation results of VAN and TAN are shown along with the fused estimation result. It can be seen in the figure that the effectiveness of the fusion algorithm is apparent. While the vertical accuracy of VAN decreases, in the fused result, TAN is more dominant in the vertical channel. In the horizontal, however, the opposite is true. This shows the GA fusion rule is able to emphasize the advantages of the individual sensor modalities.

Fig. 5.16 shows the free inertial errors vs. the fused estimation results. As can be seen from the figure, in this scenario the fusion method provides a robust navigation solution even in the absence of GNSS/GPS aiding.



(a) Trajectory side-view.



(b) Flight profile.

Figure 5.13: Scenario-3 ground truth (red) and estimated trajectory (blue).

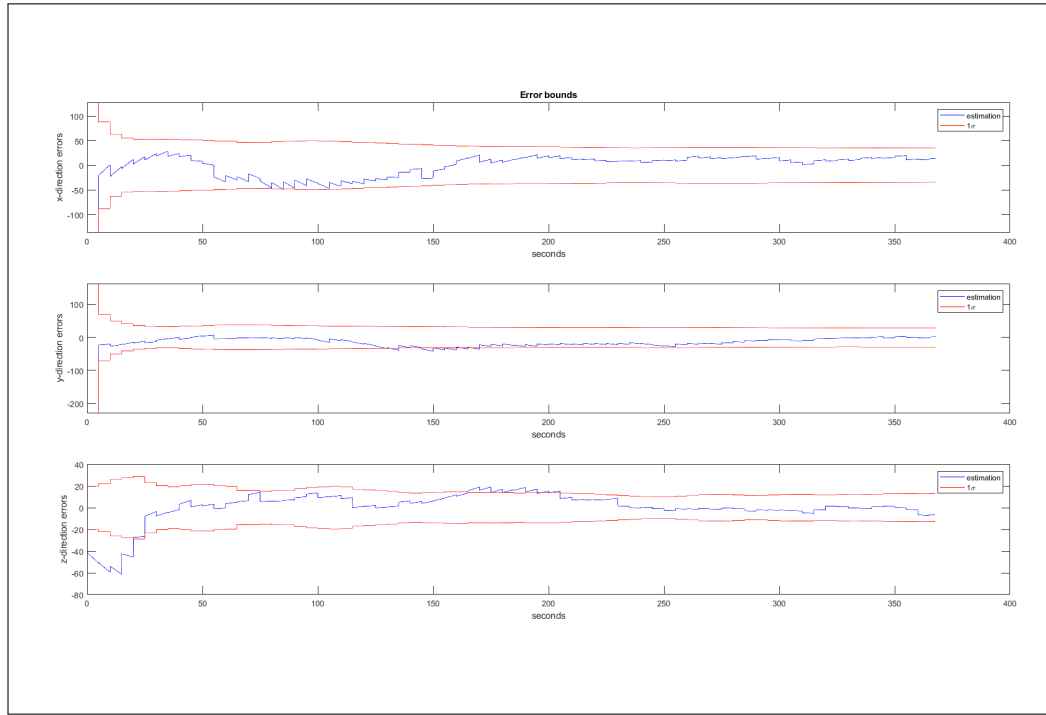


Figure 5.14: Scenario-3 error bounds.

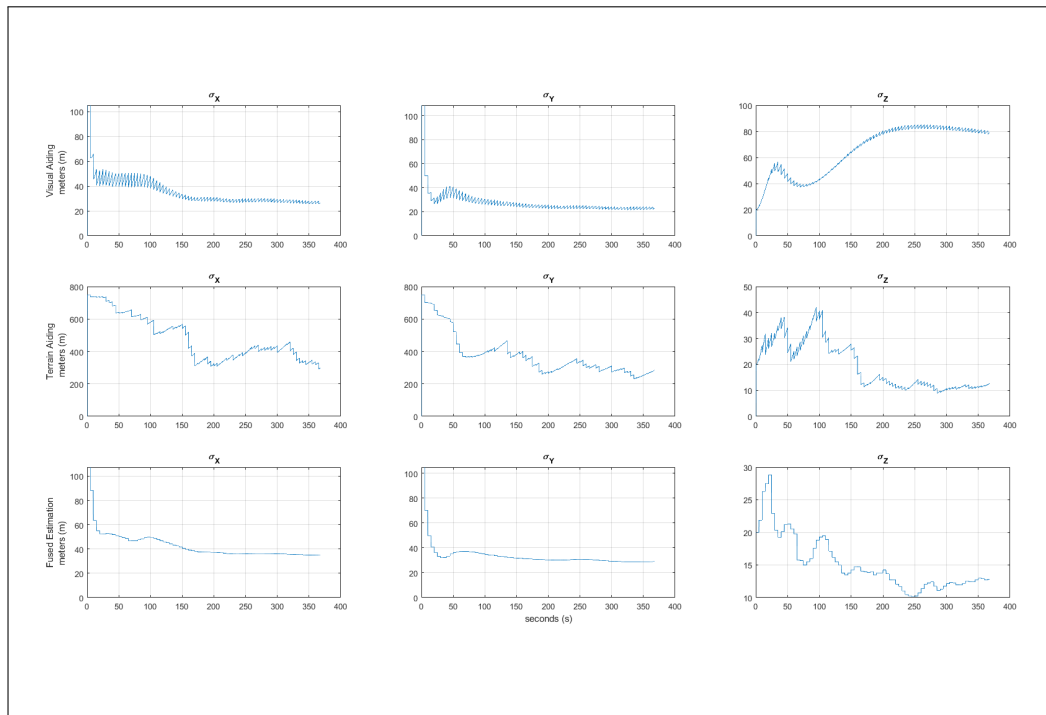


Figure 5.15: Scenario-3 VAN (top row), TAN (middle row), and GA fusion (bottom row) 3σ bounds.

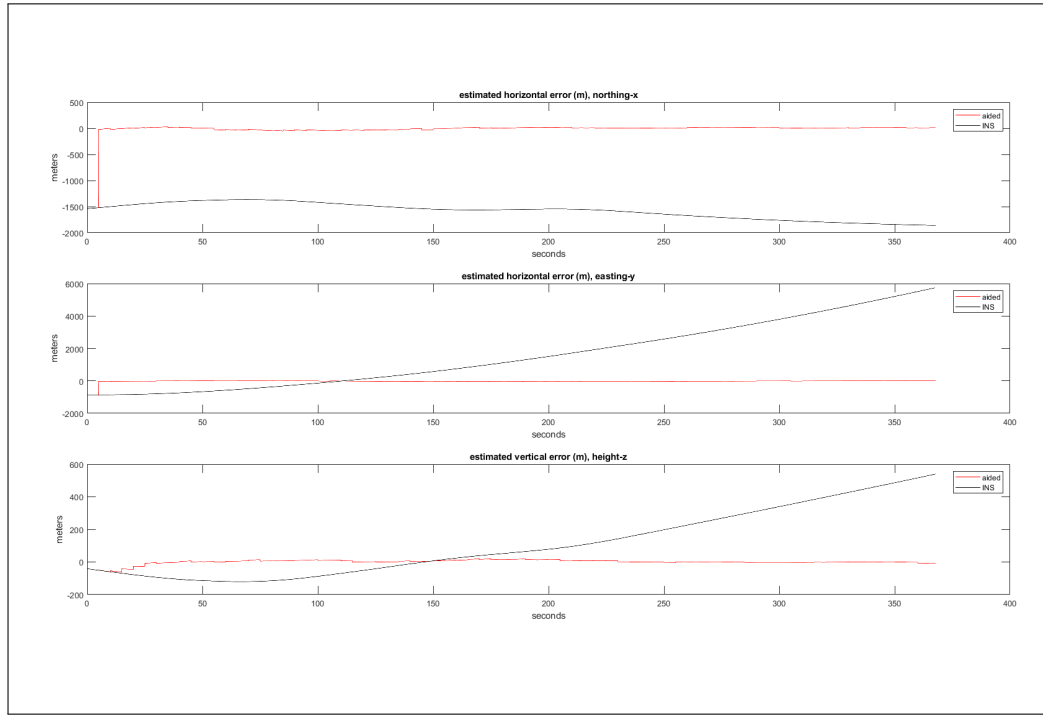


Figure 5.16: Scenario-3 free inertial error and fused estimation error.

5.5. Conclusions

In this chapter, we presented a robust navigation solution via multi-sensor fusion. The GA fusion rule is a conservative method, meaning that it is able to eliminate the need to explicitly calculate the common information between sensor nodes, which results from the inertial error propagation process. Simulation results show that the proposed fusion scheme is effective when GNSS signals are not available for aiding and can be used for assured navigation concepts.

The proposed fusion architecture is scalable. By fusing posterior level information, it is ensured that even if one of the fusing systems is not available, the integrity of the solution, however less accurate, can be maintained. Also, the addition of new sensor suites is easy to integrate into the existing structure. Since the fusion is done at the estimation level, the data structure of the existing and newly added systems would be the same.

Utilization of the GA method provides efficient computational cost. Not having to explicitly calculate the inter-sensor cross-correlations reduces the computational requirements of the fusion process. The trade-off is, however, the sub-optimal fusion result. Regardless, the simulation results show that even if it is sub-optimal, the GA fusion rule is able to provide good navigation accuracy.

6. CONCLUSIONS AND FUTURE WORKS

6.1. Conclusions

This dissertation explores the application of Bayesian unified data fusion to state estimation problems across different abstraction levels as delineated by the Joint Directors of Laboratories (JDL) model. The study offers an extensive analysis of various data fusion methodologies currently utilized in the realm of statistical estimation literature, specifically within contexts such as target tracking and navigation. The research herein proposed introduces innovative fusion algorithms addressing issues from level 1 to level 5, as defined by the JDL model. The effectiveness of these algorithms is evaluated through a series of comprehensive simulations. Throughout the algorithm development process, we employ the Random Finite Set (RFS) framework as a means to represent a spectrum of uncertainties, including those related to measurements and object ambiguities.

The first algorithm introduced establishes novel dynamic and measurement models specifically for cluster-type targets, encompassing the level 1 and level 2 fusion. The process of generating these measurements is ambiguous and unknown, as are the measurements observed, which are presented in the form of point clouds. This particular issue is also identified as *dynamic clustering*. It encompasses the detection and classification of observed measurements over time into an unknown quantity of clusters of undetermined sizes.

For cluster-type targets, this study proposes the use of the filter called Cluster-Target Probability Hypothesis Density (CT-PHD). This novel algorithm utilizes the Dirichlet Process Mixture Model (DPMM) to partition the measurement clouds into an unknown number of target clusters. Subsequently, the algorithm propagates the statistics of these clusters—such as center and shape—through the Probability Hypothesis Density (PHD) equations. The dynamic and measurement models are formulated using generalized likelihood functions and probability-generating functionals (p.g.fl.s). Furthermore, this filter relies on the Sequential Monte Carlo (SMC) implementation. The proposed filtering algorithm is then applied to a multi-target tracking scenario. In this context, real target trajectories are employed, with measurements derived from intercepted communication signals by the TDOA algorithm. This scenario encompasses ten targets, each moving in distinct

directions. Simulation outcomes effectively demonstrate the proposed algorithm's proficiency in identifying target clusters. These clusters' shape is contingent on the relative position of the target in relation to the receiving sensors. The results thus highlight the practicality and efficiency of the proposed algorithm in real-world multi-target tracking scenarios.

Level 3 and level 4 fusion are investigated through the sensor management problem in which the next position of a moving sensor platform is estimated in a multi-target surveillance scenario. We put forth a novel Random Finite Set (RFS)-based information divergence functional, leveraging the statistical Hölder divergence. We further extend this functional to RFS densities, specifically focusing on Poisson Point Processes (PPPs), and we denote this as the Extended Hölder Divergence (EHD). We present mathematical expressions for both the Sequential Monte Carlo (SMC) and Gaussian Mixture Model (GMM) formulations of the EHD and apply these formulations to a sensor management problem. Simulation results imply the efficacy of the proposed EHD functional in predicting the subsequent position of the sensor platform to optimize the observation of multiple targets. Furthermore, these results also suggest the possibility of adjusting the EHD parameters to adapt the management algorithm to fulfill different, and possibly conflicting, objectives. These include improved localization of multiple targets and a more accurate estimation of the number of targets present in the scene. Lastly, comparative analysis reveals that the proposed algorithm consistently outperforms similar techniques, such as the Cauchy-Schwarz divergence (CSD) and Rènyi divergence, thereby validating its superior performance and adaptability in complex sensor management contexts.

In terms of level 5 fusion, this study delves into the application of estimation fusion techniques—specifically posterior fusion—within the context of a multi-sensor navigation problem. This approach incorporates high-level information derived from an array of sensors, including inertial sensors, camera imaging, and radar altimeters. The primary objective is to integrate these diverse data streams to achieve superior platform localization. The integration of such disparate sources of information allows for a more accurate, robust, and reliable estimation of the platform's location, demonstrating the potential of level 5 fusion techniques in enhancing navigational accuracy. We adopt the Chernoff fusion methodology, originally derived from the target tracking literature, with the explicit objective of enhancing the accuracy of position estimation. The integration process is facilitated via a distributed architecture wherein each sensor unit independently processes its own measurements

and subsequently transmits its posterior estimate to a fusion node. At this node, these posterior estimates are effectively fused together. These posteriors are characterized by error estimates, prompting the fusion algorithm to utilize error states rather than the typically employed absolute states. To the best of our knowledge, this represents a novel application of these techniques in this context, as such an application has not been previously documented within the existing literature. This approach, therefore, not only improves estimation accuracy but also contributes to the body of knowledge in this field.

6.2. Future Works

In this section, possible future research problems are presented, based on the findings of this research.

- The parameters of the Extended Hölder Divergence (EHD) are manually adjusted in the present study. However, future research could explore an optimization technique to autonomously determine these parameter values, contingent upon the current state of the scene. Such a technique might effectively ascertain which objective aspect—such as target localization or cardinality estimation—should be prioritized. This approach would be particularly beneficial in dynamic scenarios.
- The findings of this research suggest that sensor trajectories predicted in a temporally-myopic scheme (i.e., single-step look ahead) exhibit a zig-zagging pattern. An extension to the current technique could involve a model-myopic method that looks multiple steps ahead in time, replacing the temporally-myopic approach. Despite its potential computational challenges, such a method may generate smoother trajectories for the sensor platform, contributing significantly to the field of sensor management.
- The application of posterior fusion for navigation could be further enhanced by incorporating integrity monitoring algorithms, such as Receiver Autonomous Integrity Monitoring (RAIM), when optimizing fusion weights. This contrasts with blind optimization that exclusively depends on current estimation covariances. This weight optimization scheme would further exploit the

information functionals utilized in this research, offering a valuable contribution to both fusion and navigation literature.

- A potential avenue for future study could involve leveraging the information functional-based sensor management concept presented in this research for navigation purposes. This application could be particularly beneficial when the platform moves between waypoints, with the path determined by a sensor management algorithm. This could ensure the selection of the most informative path—such as traversing a feature-rich terrain—to maintain navigation accuracy at all times. This would be particularly useful in long-duration missions where satellite signals are unavailable for extended periods.
- In the current work, when modeling the cluster-type targets, we assume that the cluster size and shape parameters are independent of target kinematics. However, certain measurement generation processes may find the size and shape of generated measurement clouds dependent on the target-sensor geometry. Therefore, considering the target's position relative to the sensors when modeling the cluster targets' time evolution would be a beneficial contribution.
- This study employs a model-based clustering scheme based on the Bayesian nonparametric framework. Alternatively, a data-driven method for dynamically clustering measurements into targets and computing their statistics could offer valuable insights. Given the rapid advancements in computational power, deep neural network structures like Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks could be utilized to process time-series data in such a context.

REFERENCES

- [1] Castanedo F., (2013), “A Review of Data Fusion Techniques”, The Scientific World Journal, 2013, 704504.publisher: Hindawi Publishing Corporation.
- [2] Durrant-Whyte H.F., (1988), “Sensor Models and Multisensor Integration”, The International Journal of Robotics Research, 7(6), 97–113.
- [3] Dasarthy B.V., (1997), “Sensor fusion potential exploitation-innovative architectures and illustrative applications”, Proceedings of the IEEE, 85(1), 24–38.
- [4] Luo R.C., Yih C.C., Su K.L., (2002), “Multisensor fusion and integration: approaches, applications, and future research directions”, IEEE Sensors Journal, 2(2), 107–119.
- [5] Hall D.L., Llinas J., (1997), “An introduction to multisensor data fusion”, Proceedings of the IEEE, 85(1), 6–23.
- [6] Hall D., Chong C.Y., Llinas J., Liggins M., (2012), “Distributed Data Fusion for Network-Centric Operations”, 1st, CRC Press.
- [7] Mullane J., Vo B.N., Adams M.D., Vo B.T., (2011), “Random finite sets for robot mapping and SLAM: new concepts in autonomous robotic map representations”, 1st, Springer-Verlag.oCLC: ocn741541294.
- [8] Zhou T., Yan Y., (2014), “Video target tracking based on mean shift algorithm with Kalman filter”, 2014 10th International Conference on Natural Computation (ICNC), 980–984, Xiamen, China, December.
- [9] Hammarberg B., Forster C., Torebjork E., (2002), “Parameter estimation of human nerve C-fibers using matched filtering and multiple hypothesis tracking”, IEEE Transactions on Biomedical Engineering, 49(4), 329–336.
- [10] Goodman I.R., Mahler R.P.S., Nguyen H.T., (1997), “Mathematics of Data Fusion”, 1st, Springer Netherlands.
- [11] Dou Y., Ran C., Gao Y., (2016), “Weighted measurement fusion Kalman estimator for multisensor descriptor system”, International Journal of Systems Science, 47(11), 2722–2732.
- [12] Naidu V.P.S., (2009), “Fusion of radar and first sensor measurements for 3d target tracking using extended kalman filter”, Defence Science Journal, 59, 175–182.
- [13] Goh S.T., Abdelkhalik O., Zekavat S.A.R., (2013), “A Weighted Measurement Fusion Kalman Filter implementation for UAV navigation”, Aerospace Science and Technology, 28(1), 315–323.
- [14] Grime S., Durrant-Whyte H.F., (1994), “Data fusion in decentralized sensor networks”, Control Engineering Practice, 2(5), 849 – 863.

- [15] Zhang C., Wang H., (2010), “Decentralized Multi-sensor Data Fusion Algorithm Using Information Filter”, 2010 International Conference on Measuring Technology and Mechatronics Automation, 890–893, Changsha, China, mar.
- [16] Uney M., Clark D.E., Julier S.J., (2013), “Distributed Fusion of PHD Filters Via Exponential Mixture Densities”, IEEE Journal of Selected Topics in Signal Processing, 7(3), 521–531.
- [17] Makarenko A., Durrant-Whyte H.F., (2006), “Decentralized bayesian algorithms for active sensor networks”, Inf. Fusion, 7, 418–433.
- [18] Rosencrantz M., Gordon G.J., Thrun S., (2012), “Decentralized sensor fusion with distributed particle filters”, CoRR, abs/1212.2493.
- [19] Bar-Shalom Y., Willett P.K., Tian X., (2011), “Tracking and Data Fusion: A Handbook of Algorithms”, 1st, YBS Publishing.
- [20] Blackman S.S., Popoli R., (1999), “Design and Analysis of Modern Tracking Systems”, 1st, Artech House.
- [21] Vo B.T., (2008), “Random Finite Sets in Multi-Object Filtering” [Ph.D.]. The University of Western Australia.
- [22] Fantacci C., Vo B.N., Vo B.T., Battistelli G., Chisci L., (2015), “Consensus Labeled Random Finite Set Filtering for Distributed Multi-Object Tracking”, arXiv:1501.01579 [cs, stat], arXiv: 1501.01579.
- [23] Fantacci C., Vo B.N., Vo B.T., Battistelli G., Chisci L., (2018), “Robust fusion for multisensor multiobject tracking”, IEEE Signal Processing Letters, 25(5), 640–644.
- [24] Kirubarajan T., Wang H., Bar-Shalom Y., Pattipati K.R., (2001), “Efficient multisensor fusion using multidimensional data association”, IEEE Transactions on Aerospace and Electronic Systems, 37(2), 386–400.
- [25] Rezatofighi S.H., Milan A., Zhang Z., Shi Q., Dick A., Reid I., (2015), “Joint Probabilistic Data Association Revisited”, 2015 IEEE International Conference on Computer Vision (ICCV), 3047–3055, Santiago, Chile, December.
- [26] Bar-Shalom Y., Daum F., Huang J., (2009), “The probabilistic data association filter”, IEEE Control Systems Magazine, 29(6), 82–100.
- [27] Chen X., Li Y., Li Y., Yu J., Li X., (2016), “A novel probabilistic data association for target tracking in a cluttered environment”, Sensors, 16(12).
- [28] Reid D., (1979), “An algorithm for tracking multiple targets”, IEEE Transactions on Automatic Control, 24(6), 843–854.
- [29] Vo B.n., Mallick M., Bar.shalom Y., Coraluppi S., Osborne R., Mahler R., Vo B.t., (2015), “Multitarget Tracking”, Wiley Encyclopedia of Electrical and Electronics Engineering, 1–15, Hoboken, NJ, USA, Sep..
- [30] Vo B.T., Vo B.N., Hoseinnezhad R., Mahler R.P.S., (2013), “Robust Multi-Bernoulli Filtering”, IEEE Journal of Selected Topics in Signal Processing, 7(3), 399–409.

- [31] Fatemi M., Granstrom K., Svensson L., Ruiz F.J.R., Hammarstrand L., (2017), “Poisson Multi-Bernoulli Mapping Using Gibbs Sampling”, *IEEE Transactions on Signal Processing*, 65(11), 2814–2827.
- [32] Mahler R., (2017), “Measurement-to-track association and finite-set statistics”, *arXiv:1701.07078v1 [stat.ME]*, .
- [33] Mahler R.P.S., (2014), “Advances in statistical multisource-multitarget information fusion”, 1st, Artech House.oCLC: ocn876671549.
- [34] Li P., wei Ge H., long Yang J., Wang W., (2018), “Modified gaussian inverse wishart phd filter for tracking multiple non-ellipsoidal extended targets”, *Signal Processing*, 150, 191–203.
- [35] Mahler R., (2009), “Phd filters for nonstandard targets, i: Extended targets”, 2009 12th International Conference on Information Fusion, 915-921, Seattle, WA, USA, August.
- [36] Orguner U., (2012), “A Variational Measurement Update for Extended Target Tracking With Random Matrices”, *IEEE Transactions on Signal Processing*, 60(7), 3827–3834.
- [37] Titterton D., Weston J., (2004), “Strapdown inertial navigation technology”, 2nd, Institution of Engineering and Technology.
- [38] Xue X., Huang S., Wei D., Xie J., (2022), “Multiradar Joint Tracking of Cluster Targets Based on Graph-LSTMs”, *Journal of Sensors*, 2022, 1–20.
- [39] Pinto J., Hess G., Ljungbergh W., Xia Y., Wymeersch H., Svensson L., (2022), “Can deep learning be applied to model-based multi-object tracking?”, *arXiv.org*, .
- [40] Li L., Dai C., Xia Y., Svensson L., (2023), “Deep Fusion of Multi-Object Densities Using Transformer”, *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5, Rhodes Island, Greece, Jun..
- [41] Li T., Song Y., Fan H., (2023), “From target tracking to targeting track: A data-driven yet analytical approach to joint target detection and tracking”, *Signal Processing*, 205, 108883.
- [42] Mahler R.P.S., (2002), “Extended first-order Bayes filter for force aggregation”, *Signal and Data Processing of Small Targets 2002*, 196 – 207, Orlando, FL, USA, July.
- [43] Mahler R., (2020), “The Pairwise-Markov Bernoulli Filter”, *IEEE Access*, 8, 168229–168245.
- [44] Granstrom K., Baum M., Reuter S., (2017), “Extended Object Tracking: Introduction, Overview and Applications”, *arXiv:1604.00970 [cs, eess]*, .arXiv: 1604.00970.
- [45] Granström K., Lundquist C., (2013), “On the use of multiple measurement models for extended target tracking”, *Proceedings of the 16th International Conference on Information Fusion*, 1534–1541, İstanbul, Turkey, July.

- [46] Baum M., Hanebeck U.D., (2011), “Shape tracking of extended objects and group targets with star-convex RHM’s”, 14th International Conference on Information Fusion, 1–8, Chicago, IL, USA, July.
- [47] Cheng X., Ji H., Zhang Y., (2019), “Improved Box Particle CPHD Algorithm for Group Target Tracking”, 2019 International Conference on Control, Automation and Information Sciences (ICCAIS), 1–7, Chengdu, China, October.
- [48] Wang Y., Hu G.p., Zhou H., (2016), “Group Targets Tracking Using Multiple Models GGIW-CPHD Based on Best-Fitting Gaussian Approximation and Strong Tracking Filter”, Journal of Sensors, 2016, 7294907.
- [49] Campbell T., Kulis B., How J., (2017), “Dynamic Clustering Algorithms via Small-Variance Analysis of Markov Chain Mixture Models”, arXiv:1707.08493 [stat], .arXiv: 1707.08493.
- [50] Topkaya I.S., Erdogan H., Porikli F., (2016), “Tracklet clustering for robust multiple object tracking using distance dependent Chinese restaurant processes”, Signal, Image and Video Processing, 10(5), 795–802.
- [51] Narula K., Garai T., Mukhopadhyay S., (2005), “An Effective Guidance Scheme for Passive Tracking Using a Quantized Bearing-only Sensor”, 2005 Annual IEEE India Conference - Indicon, 299–303, Chennai, India, September.
- [52] Wang Y., Wang X., Cui N., (2020), “Quantized feedback particle filter for unmanned aerial vehicles tracking with quantized measurements”, Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, p. 095441002094268.
- [53] Aoki E.H., Bagchi A., Mandal P., Boers Y., (2011), “A theoretical look at information-driven sensor management criteria”, 14th International Conference on Information Fusion, 1–8, Chicago, IL, USA, July.
- [54] Ferri G., Munafo A., LePage K.D., (2018), “An Autonomous Underwater Vehicle Data-Driven Control Strategy for Target Tracking”, IEEE Journal of Oceanic Engineering, 43(2), 323–343.
- [55] Li W., Han C., (2018), “Dual Sensor Control Scheme for Multi-Target Tracking”, Sensors (Basel, Switzerland), 18(5).
- [56] Gostar A.K., Hoseinnezhad R., Bab.Hadiashar A., (2016), “Multi-Bernoulli sensor-selection for multi-target tracking with unknown clutter and detection profiles”, Signal Processing, 119, 28–42.
- [57] Aoki E.H., Bagchi A., Mandal P., Boers Y., (2011), “On the “near-universal proxy” argument for theoretical justification of information-driven sensor management”, 2011 IEEE Statistical Signal Processing Workshop (SSP), 245–248, Nice, France, June.
- [58] Battistelli G., Chisci L., Fantacci C., Farina A., Graziano A., (2013), “Consensus CPHD Filter for Distributed Multitarget Tracking”, IEEE Journal of Selected Topics in Signal Processing, 7(3), 508–520.
- [59] Gao L., Battistelli G., Chisci L., (2022), “Fusion of Labeled RFS Densities With Different Fields of View”, IEEE Transactions on Aerospace and Electronic Systems, 1–16.

- [60] Üney M., Houssineau J., Delande E., Julier S.J., Clark D.E., (2018), “Fusion of finite set distributions: Pointwise consistency and global cardinality”, arXiv:1802.06220 [cs, eess, math], .arXiv: 1802.06220.
- [61] Julier S.J., (2008), “Fusion without independence”, IET Seminar on Target Tracking and Data Fusion: Algorithms and Applications, 1–4, Birmingham, UK, April.
- [62] Battistelli G., Chisci L., Morrocchi S., Papi F., (2011), “An Information-Theoretic Approach to Distributed State Estimation”, IFAC Proceedings Volumes, 44(1), 12477–12482.
- [63] Bayes T., (1763), “An essay towards solving a problem in the doctrine of chances”, Philosophical Transactions, 53, 370–418.
- [64] Kalman R.E., (1960), “A New Approach to Linear Filtering and Prediction Problems”, Transactions of the ASME–Journal of Basic Engineering, 82(Series D), 35–45.
- [65] Aeberhard M., Schlichtharle S., Kaempchen N., Bertram T., (2012), “Track-to-Track Fusion With Asynchronous Sensors Using Information Matrix Fusion for Surround Environment Perception”, IEEE Transactions on Intelligent Transportation Systems, 13(4), 1717–1726.
- [66] Yaakov Bar-Shalom, X. Rong Li, Thiagalingam Kirubarajan, (2001), “Estimation with Applications to Tracking & Navigation”, 1. Edt, Wiley-Interscience.
- [67] Ristic B., Arulampalam S., Gordon N.J., (2004), “Beyond the kalman filter: Particle filters for tracking applications”, 1st, Artech House.
- [68] Sorenson H.W., (1974), “On the development of practical nonlinear filters”, Information Sciences, 7, 253 – 270.
- [69] Nam Trung Pham, Weimin Huang, Ong S.H., (2007), “Multiple sensor multiple object tracking with GMPHD filter”, 2007 10th International Conference on Information Fusion, 1–7, Quebec, QC, Canada, July.
- [70] Vo B., Ma W., (2006), “The Gaussian Mixture Probability Hypothesis Density Filter”, IEEE Transactions on Signal Processing, 54(11), 4091–4104.
- [71] Veetil D.R.A., Chakravorty S., (2016), “Particle Gaussian Mixture (PGM) Filters”, CoRR, abs/1603.04510.
- [72] Blom H.A.P., Bar.Shalom Y., (1988), “The interacting multiple model algorithm for systems with Markovian switching coefficients”, IEEE Transactions on Automatic Control, 33(8), 780–783.
- [73] Mazor E., Averbuch A., Bar.Shalom Y., Dayan J., (1998), “Interacting multiple model methods in target tracking: a survey”, IEEE Transactions on Aerospace and Electronic Systems, 34(1), 103–123.
- [74] Julier S.J., Uhlmann J.K., (1997), “A New Extension of the Kalman Filter to Nonlinear Systems”, Proceedings Volume 3068, Signal Processing, Sensor Fusion, and Target Recognition VI, 182–193, Orlando, FL, USA, July.

- [75] Wan E.A., Van.Der.Merwe R., (2000), "The unscented Kalman filter for nonlinear estimation", Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373), 153–158, Lake Louise, Alta., Canada, October.
- [76] Xu Y., Liping L., (2004), "Single observer bearings-only tracking with the unscented Kalman filter", 2004 International Conference on Communications, Circuits and Systems (IEEE Cat. No.04EX914), 901–905 Vol.2, Chengdu, China, Jun..
- [77] Yi W., Morelande M.R., Kong L., Yang J., (2013), "A Computationally Efficient Particle Filter for Multitarget Tracking Using an Independence Approximation", IEEE Transactions on Signal Processing, 61(4), 843–856.
- [78] Park S., Hwang J., Rou K., Kim E., (2007), "A New Particle Filter Inspired by Biological Evolution: Genetic Filter", International Journal of Electronics and Communication Engineering, 1(9), 5.
- [79] Charalampidis A.C., Papavassilopoulos G.P., (2013), "Improved auxiliary and Unscented Particle Filter variants", 52nd IEEE Conference on Decision and Control, 7040–7046, Firenze, Italy, Dec..
- [80] Liang.qun L., Wei.xin X., (2014), "Intuitionistic fuzzy joint probabilistic data association filter and its application to multitarget tracking", Signal Processing, 96, 433–444.
- [81] Habtemariam B., Tharmarasa R., Thayaparan T., Mallick M., Kirubarajan T., (2013), "A Multiple-Detection Joint Probabilistic Data Association Filter", IEEE Journal of Selected Topics in Signal Processing, 7(3), 461–471.
- [82] Amditis A., Thomaidis G., Maroudis P., Lytrivis P., Karaseitanidis G., (2012), "Multiple Hypothesis Tracking Implementation", Laser Scanner Technology, p. 23.
- [83] Kosuge Y., Tachibana Y., Tsujimichi S., (1999), "A track-oriented multiple hypothesis multitarget tracking algorithm", Electronics and Communications in Japan (Part I: Communications), 82(12), 84–92.
- [84] Sun J., Li Q., Zhang X., Sun W., (2017), "An Efficient Implementation of Track-Oriented Multiple Hypothesis Tracker Using Graphical Model Approaches", Mathematical Problems in Engineering, 2017, 1–11.
- [85] Vo B.n., Mallick M., Bar.shalom Y., Coraluppi S., Osborne.III R., Mahler R., Vo B.t., (2015), "Multitarget tracking", Wiley Encyclopedia of Electrical and Electronics Engineering, 1–15.
- [86] Mahler R.P.S., (2007), "Statistical multisource-multitarget information fusion", , Artech House.oCLC: ocn122257615.
- [87] Mahler R.P.S., (2004), ""Statistics 101" for multisensor, multitarget data fusion", IEEE Aerospace and Electronic Systems Magazine, 19(1), 53–64.
- [88] Mahler R., (2001), "Random Set Theory for Target Tracking and Identification", 1st, CRC Press.

- [89] Mahler R., (2013), ““Statistics 102” for Multisource-Multitarget Detection and Tracking”, *IEEE Journal of Selected Topics in Signal Processing*, 7(3), 376–389.
- [90] Liu Z., Chen S., Wu H., He R., Hao L., (2018), “A Student’s t Mixture Probability Hypothesis Density Filter for Multi-Target Tracking with Outliers”, *Sensors*, 18(4), 1095.
- [91] Sithiravel R., McDonald M., Balaji B., Kirubarajan T., (2016), “Multiple model spline probability hypothesis density filter”, *IEEE Transactions on Aerospace and Electronic Systems*, 52(3), 1210–1226.
- [92] Maher R., (2007), “A survey of PHD filter and CPHD filter implementations”, *Signal Processing, Sensor Fusion, and Target Recognition XVI*, 65670O, Orlando, Florida, United States, Apr..
- [93] Ristic B., Clark D., Vo B.N., Vo B.T., (2012), “Adaptive Target Birth Intensity for PHD and CPHD Filters”, *IEEE Transactions on Aerospace and Electronic Systems*, 48(2), 1656–1668.
- [94] Balaji B., Kirubarajan T., Baser E., Efe M., (2016), “Improved multi-target multi-Bernoulli filter with modelling of spurious targets”, *IET Radar, Sonar & Navigation*, 10(2), 285–298.
- [95] Vo B., Vo B., (2013), “Labeled Random Finite Sets and Multi-Object Conjugate Priors”, *IEEE Transactions on Signal Processing*, 61(13), 3460–3475.
- [96] Beard M., Vo B.T., Vo B.N., Arulampalam S., (2017), “Void Probabilities and Cauchy–Schwarz Divergence for Generalized Labeled Multi-Bernoulli Models”, *IEEE Transactions on Signal Processing*, 65(19), 5047–5061.
- [97] Lin S., Vo B.T., Nordholm S.E., (2016), “Measurement driven birth model for the generalized labeled multi-Bernoulli filter”, *Control, Automation and Information Sciences (ICCAIS)*, 2016 International Conference on, 94–99, Ansan, Korea (South), October.
- [98] Beard M.A., (2016), “Estimation and Control of Multi-object Systems with High-fidelity Sensor Models” [Ph.D.]. The University of Western Australia.
- [99] Mahler R.P.S., (2003), “Multitarget Bayes filtering via first-order multitarget moments”, *IEEE Transactions on Aerospace and Electronic Systems*, 39(4), 1152–1178.
- [100] Mahler R., (2022), “Mathematical Representation of Multitarget Systems”, *arXiv:2203.10972 [math]*, .
- [101] Reineking T., (2014), “Belief Functions: Theory and Algorithms” [Ph.D. Dissertation]. University of Bremen.
- [102] Mahler R., (2019), “Exact Closed-Form Multitarget Bayes Filters”, *Sensors*, 19(12), 2818.
- [103] Granstrom K., Fatemi M., Svensson L., (2019), “Poisson multi-Bernoulli conjugate prior for multiple extended object filtering”, *arXiv:1605.06311 [cs, stat]*, .

- [104] Ba-Ngu Vo, Singh S., Doucet A., (2003), “Sequential monte carlo implementation of the phd filter for multi-target tracking”, Sixth International Conference of Information Fusion, 2003. Proceedings of the, 792–799, Cairns, QLD, Australia, Jul..
- [105] Özer E., (2022), “SUALTI ÇOK SENSÖRLÜ ÇOKLU HEDEF TAKİBİ” [PhD Thesis]. GEBZE TECHNICAL UNIVERSITY.
- [106] Mahler R., (2009), “CPHD and PHD filters for unknown backgrounds II: multitarget filtering in dynamic clutter”, Proc. SPIE 7330, Sensors and Systems for Space Applications III, 73300L, Orlando, Florida, USA, May.
- [107] Garcia.Fernandez A.F., Williams J.L., Granstrom K., Svensson L., (2018), “Poisson multi-Bernoulli mixture filter: direct derivation and implementation”, IEEE Transactions on Aerospace and Electronic Systems, 1–1.
- [108] Llinas M.L.D.H.J., (2008), “Handbook of Multisensor Data Fusion: Theory and Practice”, 2nd Edition, CRC Press.
- [109] Battistelli G., Chisci L., (2014), “Kullback–Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability”, Automatica, 50(3), 707–718.
- [110] Mahler R., (2015), “A brief survey of advances in random-set fusion”, 2015 International Conference on Control, Automation and Information Sciences (ICCAIS), 62–67, Changshu, China, Oct..
- [111] Gunay M., Orguner U., Demirekler M., (2016), “Chernoff fusion of Gaussian mixtures based on sigma-point approximation”, IEEE Transactions on Aerospace and Electronic Systems, 52(6), 2732–2746.
- [112] Mahler R.P.S., (2000), “Optimal/robust distributed data fusion: a unified approach”, Distributed Data Fusion for Network-Centric Operations, 128–138, Orlando, FL, Aug..
- [113] Li T., Fan H., García J., Corchado J.M., (2019), “Second-order statistics analysis and comparison between arithmetic and geometric average fusion: Application to multi-sensor target tracking”, Information Fusion, 51, 233–243.
- [114] Schuhmacher D., Vo B.T., Vo B.N., (2008), “A Consistent Metric for Performance Evaluation of Multi-Object Filters”, IEEE Transactions on Signal Processing, 56(8), 3447–3457.
- [115] Ristic B., Vo B.N., Clark D., (2010), “Performance evaluation of multi-target tracking using the ospa metric”, 2010 13th International Conference on Information Fusion, 1-7, Edinburgh, UK, February.
- [116] Rogers R.M., (2007), “Applied mathematics in integrated navigation systems, third edition”, 3rd, American Institute of Aeronautics and Astronautics, Inc..
- [117] Groves P.D., (2015), “Navigation using inertial sensors [Tutorial]”, IEEE Aerosp. Electron. Syst. Mag., 30(2), 42–69.
- [118] Mahler R., (2011), “Measurement-to-track association for nontraditional measurements”, Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on, 1–8, Chicago, IL, USA, July.

- [119] Hoseinnezhad R., Vo B.N., Vo B.T., Suter D., (2012), “Visual tracking of numerous targets via multi-bernoulli filtering of image data”, *Pattern Recognition*, 45(10), 3625–3635.
- [120] Nannuru S., Coates M., Mahler R., (2013), “Computationally-Tractable Approximate PHD and CPHD Filters for Superpositional Sensors”, *IEEE Journal of Selected Topics in Signal Processing*, 7(3), 410–420.
- [121] Mahler R., El.Fallah A., (2012), “The random set approach to nontraditional measurements is rigorously Bayesian”, *Signal Processing, Sensor Fusion, and Target Recognition XXI*, 83920D, Baltimore, Maryland, U.S., May.
- [122] Balakumar B., Sinha A., Kirubarajan T., Reilly J.P., (2005), “Phd filtering for tracking an unknown number of sources using an array of sensors”, *IEEE/SP 13th Workshop on Statistical Signal Processing*, 2005, 43-48, Bordeaux, France, July.
- [123] Hauschildt D., (2011), “Gaussian mixture implementation of the cardinalized probability hypothesis density filter for superpositional sensors”, *2011 International Conference on Indoor Positioning and Indoor Navigation*, 1-8, Guimaraes, Portugal, September.
- [124] Thouin F., Nannuru S., Coates M., (2011), “Multi-target tracking for measurement models with additive contributions”, *14th International Conference on Information Fusion*, 1-8, Chicago, IL, USA, July.
- [125] Vo B.N., Vo B.T., Pham N.T., Suter D., (2009), “Bayesian multi-object estimation from image observations”, *2009 12th International Conference on Information Fusion*, 890-898, Seattle, WA, USA, July.
- [126] Gilholm K., Godsill S., Maskell S., Salmond D., (2005), “Poisson models for extended target and group tracking”, *SIGNAL AND DATA PROCESSING OF SMALL TARGETS 2005*, 59130R, San Diego, California, USA, Aug..
- [127] Ristic B., Vo B.T., Vo B.N., Farina A., (2013), “A Tutorial on Bernoulli Filters: Theory, Implementation and Applications”, *IEEE Transactions on Signal Processing*, 61(13), 3406–3430.
- [128] Gilholm K., Salmond D., (2005), “Spatial distribution model for tracking extended objects”, *Sonar and Navigation IEE Proceedings - Radar*, 152(5), 364–371.
- [129] Ristic B., Sherrah J., (2013), “Bernoulli filter for detection and tracking of an extended object in clutter”, *2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 306–311, Melbourne, VIC, Australia, Apr..
- [130] Granström K., Natale A., Braca P., Ludeno G., Serafino F., (2015), “Gamma Gaussian Inverse Wishart Probability Hypothesis Density for Extended Target Tracking Using X-Band Marine Radar Data”, *IEEE Transactions on Geoscience and Remote Sensing*, 53(12), 6617–6631.
- [131] Vivone G., Braca P., Granstrom K., Willett P., (2016), “Multistatic Bayesian extended target tracking”, *IEEE Transactions on Aerospace and Electronic Systems*, 52(6), 2626–2643.

- [132] Waxman M.J., Drummond O.E., (2004), “A bibliography of cluster (group) tracking”, Signal and Data Processing of Small Targets 2004, 551 – 560, Orlando, Florida, United States, August.
- [133] Gordon N.T., Salmond D.J., Fisher D.J., (1997), “Bayesian target tracking after group pattern distortion”, Signal and Data Processing of Small Targets 1997, 238 – 248, San Diego, CA, United States, October.
- [134] Lian F., Han C.Z., Liu W.F., Yan X.X., Zhou H.Y., (2010), “Sequential Monte Carlo implementation and state extraction of the group probability hypothesis density filter for partly unresolvable group targets-tracking problem”, IET Radar, Sonar & Navigation, 4(5), 685–702(17).
- [135] Lan J., Li X.R., (2014), “Tracking of Maneuvering Non-Ellipsoidal Extended Object or Target Group Using Random Matrix”, IEEE Transactions on Signal Processing, 62(9), 2450–2463.
- [136] Feldmann M., Franken D., Koch W., (2011), “Tracking of Extended Objects and Group Targets Using Random Matrices”, IEEE Transactions on Signal Processing, 59(4), 1409–1420.
- [137] Mahler R.P.S., (2003), “Bayesian cluster detection and tracking using a generalized Cheeseman approach”, Proceedings of SPIE Conference, 334–345, Orlando, FL, Aug..
- [138] Mahler R., (2009), “CPHD and PHD filters for unknown backgrounds I: dynamic data clustering”, Proc. SPIE 7330, Sensors and Systems for Space Applications III, 73300K, Orlando, Florida, USA, May.
- [139] CHEESEMAN P., KELLY J., SELF M., STUTZ J., TAYLOR W., FREEMAN D., (1988), “AutoClass: A Bayesian Classification System”, Machine Learning Proceedings 1988, 54 – 64, San Francisco (CA), June.
- [140] Mahler R., (2011), “General Bayes filtering of quantized measurements”, 14th International Conference on Information Fusion, 1–7, Chicago, IL, USA, Jul..
- [141] Li Y., Schofield E., Gönen M., (2019), “A tutorial on Dirichlet process mixture modeling”, Journal of Mathematical Psychology, 91, 128–144.
- [142] Moraffah B., Papandreou. Suppappola A., (2018), “Dependent Dirichlet Process Modeling and Identity Learning for Multiple Object Tracking”, 2018 52nd Asilomar Conference on Signals, Systems, and Computers, 1762–1766, Pacific Grove, CA, USA, Oct..iSSN: 1058-6393.
- [143] Teh Y.W., Jordan M.I., Beal M.J., Blei D.M., (2006), “Hierarchical Dirichlet Processes”, Journal of the American Statistical Association, 101(476), 1566–1581.
- [144] Lovejoy W.S., (1991), “A survey of algorithmic methods for partially observed Markov decision processes”, Annals of Operations Research, 28(1), 47–65.
- [145] Xiang Y., Alahi A., Savarese S., (2015), “Learning to Track: Online Multi-object Tracking by Decision Making”, 2015 IEEE International Conference on Computer Vision (ICCV), 4705–4713, Santiago, Chile, Dec..

- [146] Wang X., Hoseinnezhad R., Gostar A.K., Rathnayake T., Xu B., Bab.Hadiashar A., (2018), “Multi-sensor control for multi-object Bayes filters”, *Signal Processing*, 142, 260–270.
- [147] Li G., Li S., Yi W., Zhou T., Kong L., (2018), “A Suboptimal Multi-Sensor Management Based on Cauchy-Schwarz Divergence for Multi-Target Tracking”, 2018 21st International Conference on Information Fusion (FUSION), 1–8, Cambridge, UK, September.
- [148] Gostar A.K., Hoseinnezhad R., Bab.Hadiashar A., (2015), “Multi-bernoulli sensor control via minimization of expected estimation errors”, *IEEE Transactions on Aerospace and Electronic Systems*, 51(3), 1762–1773.
- [149] Charlish A., Hoffmann F., (2015), “Anticipation in cognitive radar using stochastic control”, 2015 IEEE Radar Conference (RadarCon), 1692–1697, Arlington, VA, USA, May.
- [150] Gostar A.K., Hoseinnezhad R., Bab.Hadiashar A., Liu W., (2017), “Sensor-Management for Multitarget Filters via Minimization of Posterior Dispersion”, *IEEE Transactions on Aerospace and Electronic Systems*, 53(6), 2877–2884.
- [151] Mahler R.P.S., Zajic T.R., (2004), “Probabilistic objective functions for sensor management”, *SPIE Proceedings*, 233–244, Orlando, FL, Aug..
- [152] Vu T., Evans R., (2018), “Optimal Subpattern Assignment Metric for Multiple Tracks (OSPAMT Metric)”, arXiv:1808.02242 [eess], .arXiv: 1808.02242.
- [153] Gostar A.K., Hoseinnezhad R., Bab.Hadiashar A., (2013), “Robust Multi-Bernoulli Sensor Selection for Multi-Target Tracking in Sensor Networks”, *IEEE Signal Processing Letters*, 20(12), 1167–1170.
- [154] Gao L., Battistelli G., Chisci L., Wei P., (2019), “Distributed Joint Sensor Registration and Multitarget Tracking Via Sensor Network”, arXiv:1902.02523 [cs], .arXiv: 1902.02523.
- [155] Hoang H.G., Vo B.T., (2014), “Sensor management for multi-target tracking via multi-Bernoulli filtering”, *Automatica*, 50(4), 1135–1142.
- [156] Beard M., Vo B., Vo B., Arulampalam S., (2015), “Sensor control for multi-target tracking using Cauchy-Schwarz divergence”, 2015 18th International Conference on Information Fusion (Fusion), 937–944, Washington, DC, USA, Jul..
- [157] Panicker S., Gostar A.K., Bab.Hadiashar A., Hoseinnezhad R., (2019), “Recent Advances in Stochastic Sensor Control for Multi-Object Tracking”, *Sensors*, 19(17), 3790.
- [158] Monahan G.E., (1982), “A Survey of Partially Observable Markov Decision Processes: Theory, Models, and Algorithms”, *Management Science*, 28(1), 1–16.
- [159] Krishnamurthy V., (2002), “Algorithms for optimal scheduling and management of hidden Markov model sensors”, *IEEE Transactions on Signal Processing*, 50(6), 1382–1397.

- [160] Mahler R.P.S., (2003), “Multisensor-multitarget sensor management: a unified Bayesian approach”, SPIE Proceedings, 222–233, Orlando, FL, Aug..
- [161] Hero A.O., Kreucher C.M., Blatt D., (2008), “Information Theoretic Approaches to Sensor Management”, , Springer US.
- [162] Touchette H., Lloyd S., (2004), “Information-theoretic approach to the study of control systems”, *Physica A: Statistical Mechanics and its Applications*, 331(1), 140–172.
- [163] Ristic B., Vo B.N., Clark D., (2011), “A Note on the Reward Function for PHD Filters with Sensor Control”, *IEEE Transactions on Aerospace and Electronic Systems*, 47(2), 1521–1529.
- [164] Gostar A.K., Rathnayake T., Tennakoon R., Bab.Haidashar A., Hoseinnezhad R., (2018), “Non-bayesian track-before-detect using cauchy-schwarz divergence-based information fusion”, 2018 21st International Conference on Information Fusion (FUSION), 289-294, Cambridge, UK, September.
- [165] Gostar A.K., Rathnayake T., Tennakoon R., Bab.Hadiashar A., Battistelli G., Chisci L., Hoseinnezhad R., (2020), “Cooperative sensor fusion in centralized sensor networks using Cauchy–Schwarz divergence”, *Signal Processing*, 167, 107278.
- [166] Zhu Y., Mallick M., Liang S., Yan J., (2022), “Generalized labeled multi-bernoulli multi-target tracking with doppler-only measurements”, *Remote Sensing*, 14(13).
- [167] Mahler R., (2003), “Objective functions for bayesian control-theoretic sensor management, 1: multitarget first-moment approximation”, 2003 IEEE Aerospace Conference Proceedings (Cat. No.03TH8652), 4_1905–4_1923, Sky, MT, USA, Mar..iSSN: 1095-323X.
- [168] Hoang H.G., (2012), “Control of a mobile sensor for multi-target tracking using multi-target/object multi-bernoulli filter”, 2012 International Conference on Control, Automation and Information Sciences (ICCAIS), 7-12, Saigon, Vietnam, November.
- [169] Hintz K.J., McVey E.S., (1991), “Multi-process constrained estimation”, *IEEE Transactions on Systems, Man, and Cybernetics*, 21(1), 237–244.
- [170] Cui S., Datcu M., (2015), “Comparison of Kullback-Leibler divergence approximation methods between Gaussian mixture models for satellite image retrieval”, 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 3719–3722, Milan, Italy, Jul..
- [171] Hoang H.G., Vo B.N., Vo B.T., Mahler R., (2015), “The Cauchy-Schwarz divergence for Poisson point processes”, *IEEE Transactions on Information Theory*, 61(8), 4475–4485.arXiv: 1312.6224.
- [172] Nielsen F., Sun K., Marchand.Maillet S., (2017), “On Hölder Projective Divergences”, *Entropy*, 19(3), 122.
- [173] Julier S.J., (2006), “An Empirical Study into the Use of Chernoff Information for Robust, Distributed Fusion of Gaussian Mixture Models”, 2006 9th International Conference on Information Fusion, 1–8, Florence, Italy, Jul..

- [174] Ristic B., (2013), "Particle Filters for Random Set Models", , Springer New York.
- [175] Zhu Y., Liang S., Gong M., Yan J., (2022), "Decomposed pomdp optimization-based sensor management for multi-target tracking in passive multi-sensor systems", *IEEE Sensors Journal*, 22(4), 3565–3578.
- [176] Ristic B., Vo B.N., (2010), "Sensor control for multi-object state-space estimation using random finite sets", *Automatica*, 46(11), 1812–1818.
- [177] Ristic B., Clark D., Vo B.N., (2010), "Improved SMC implementation of the PHD filter", 2010 13th International Conference on Information Fusion, 1–8, Edinburgh, UK, Jul..
- [178] Aughenbaugh J.M., La.Cour B.R., (2008), "Metric selection for information theoretic sensor management", 2008 11th International Conference on Information Fusion, 1–8, Cologne, Germany, Jun..
- [179] Yang C., Kadar I., Blasch E., Bakich M., (2011), "Comparison of information theoretic divergences for sensor management", *Proc. SPIE 8050, Signal Processing, Sensor Fusion, and Target Recognition*, 80500C, Orlando, Florida, United States, May.
- [180] Farrell J.A., Silva F.O., Rahman F., Wendel J., (2022), "Inertial measurement unit error modeling tutorial: Inertial navigation system state estimation with real-time sensor calibration", *IEEE Control Systems Magazine*, 42(6), 40–66.
- [181] Petritoli E., Leccese F., Spagnolo G.S., (2020), "Inertial Navigation Systems (INS) for Drones: Position Errors Model", 2020 IEEE 7th International Workshop on Metrology for AeroSpace (MetroAeroSpace), 500–504, Pisa, Italy, Jun..
- [182] El.Sheimy N., Youssef A., (2020), "Inertial sensors technologies for navigation applications: state of the art and future trends", *Satell Navig*, 1(1), 2.
- [183] Nordlund P., Gustafsson F., (2009), "Marginalized Particle Filter for Accurate and Reliable Terrain-Aided Navigation", *IEEE Transactions on Aerospace and Electronic Systems*, 45(4), 1385–1399.
- [184] Dong Y., Wang D., Zhang L., Li Q., Wu J., (2020), "Tightly Coupled GNSS/INS Integration with Robust Sequential Kalman Filter for Accurate Vehicular Navigation", *Sensors*, 20(2), 561.
- [185] Mirzaei F.M., Roumeliotis S.I., (2008), "A Kalman Filter-Based Algorithm for IMU-Camera Calibration: Observability Analysis and Performance Evaluation", *IEEE Transactions on Robotics*, 24(5), 1143–1156.
- [186] Schumacher A., (2006), "Integration of a GPS aided Strapdown Inertial Navigation System for Land Vehicles" [Master's]. KTH Electrical Engineering.
- [187] Aslinezhad M., Malekijavan A., Abbasi P., (2020), "ANN-assisted robust GPS/INS information fusion to bridge GPS outage", *J Wireless Com Network*, 2020(1), 129.
- [188] Lu Y., Xue Z., Xia G.S., Zhang L., (2018), "A survey on vision-based UAV navigation", *Geo-spatial Information Science*, 21(1), 21–32.

- [189] Usenko V., Engel J., Stücker J., Cremers D., (2016), “Direct visual-inertial odometry with stereo cameras”, 2016 IEEE International Conference on Robotics and Automation (ICRA), 1885–1892, Stockholm, Sweden, May.
- [190] Huang G., (2019), “Visual-Inertial Navigation: A Concise Review”, 2019 International Conference on Robotics and Automation (ICRA), 9572–9582, Montreal, QC, Canada, May.
- [191] Ansar A., Daniilidis K., (2003), “Linear pose estimation from points or lines”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(5), 578–589.
- [192] Mourikis A.I., Trawny N., Roumeliotis S.I., Johnson A.E., Ansar A., Matthies L., (2009), “Vision-Aided Inertial Navigation for Spacecraft Entry, Descent, and Landing”, IEEE Transactions on Robotics, 25(2), 264–280.
- [193] Azarbayejani A., Pentland A.P., (1995), “Recursive estimation of motion, structure, and focal length”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(6), 562–575.
- [194] Munguía R., Nuño E., Aldana C.I., Urzua S., (2016), “A Visual-Aided Inertial Navigation and Mapping System”, International Journal of Advanced Robotic Systems, 13(3), 94.publisher: SAGE Publications.
- [195] Balamurugan G., Valarmathi J., Naidu V.P.S., (2016), “Survey on UAV navigation in GPS denied environments”, 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPEs), 198–204, Paralakhemundi, India, October.
- [196] Inostroza F., Adams M., Leung K., (2018), “Modeling Detection Statistics in Feature-Based Robotic Navigation for Range Sensors: Detection Statistics in Robotic Navigation”, J Inst Navig, 65(3), 297–318.
- [197] Adams M., Vo B., Mahler R., Mullane J., (2014), “SLAM Gets a PHD: New Concepts in Map Estimation”, IEEE Robotics Automation Magazine, 21(2), 26–37.conference Name: IEEE Robotics Automation Magazine.
- [198] Latif Y., Cadena C., Neira J., (2013), “Robust loop closing over time for pose graph SLAM”, The International Journal of Robotics Research, 32(14), 1611–1626.publisher: SAGE Publications Ltd STM.
- [199] Vincke B., Lambert A., Elouardi A., (2014), “Guaranteed simultaneous localization and mapping algorithm using interval analysis”, 2014 13th International Conference on Control Automation Robotics Vision (ICARCV), 1409–1414, Singapore, December.
- [200] Mullane J., Vo B.N., Adams M.D., (2010), “Rao-blackwellised phd slam”, 2010 IEEE International Conference on Robotics and Automation, 5410–5416, Anchorage, AK, USA, May.
- [201] Jianguo L., Qiang L., Qiubing L., Pengjiao Z., (2017), “Fusion of visual and inertial measurement information for unmanned aerial vehicles autonomous navigation in unknown environment”, 2017 20th International Conference on Information Fusion (Fusion), 1–6, Xi’an, China, July.

- [202] Chen C., Zhu H., Li M., You S., (2018), "A Review of Visual-Inertial Simultaneous Localization and Mapping from Filtering-Based and Optimization-Based Perspectives", *Robotics*, 7(3), 45.
- [203] Ma S., Bai X., Wang Y., Fang R., (2019), "Robust stereo visual-inertial odometry using nonlinear optimization", *Sensors*, 19(17).
- [204] Panahandeh G., Jansson M., (2014), "Vision-Aided Inertial Navigation Based on Ground Plane Feature Detection", *IEEE/ASME Transactions on Mechatronics*, 19(4), 1206–1215.
- [205] Andersen E.D., Taylor C.N., (2007), "Improving MAV pose estimation using visual information", 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, 3745–3750, San Diego, CA, USA, October.
- [206] Horn B.K.P., Schunck B.G., (1981), "Determining optical flow", *Artificial Intelligence*, 17(1), 185–203.
- [207] Lucas B.D., Kanade T., (1981), "An iterative image registration technique with an application to stereo vision", *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, 674–679, San Francisco, CA, USA, August.
- [208] Herissé B., Hamel T., Mahony R., Russotto F.X., (2012), "Landing a vtol unmanned aerial vehicle on a moving platform using optical flow", *IEEE Transactions on Robotics*, 28(1), 77–89.
- [209] Cho D.M., Tsiotras P., Zhang G., Holzinger M., (2013), "Robust feature detection, acquisition and tracking for relative navigation in space with a known target", *AIAA Guidance, Navigation, and Control (GNC) Conference*, 1–18, Boston, MA, USA, August.
- [210] Indelman V., Gurfil P., Rivlin E., Rotstein H., (2012), "Real-time vision-aided localization and navigation based on three-view geometry", *IEEE Transactions on Aerospace and Electronic Systems*, 48(3), 2239–2259.
- [211] Tardif J.P., George M., Laverne M., Kelly A., Stentz A., (2010), "A new approach to vision-aided inertial navigation", 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, 4161–4168, Taipei, Taiwan, Oct..
- [212] Hesch J.A., Kottas D.G., Bowman S.L., Roumeliotis S.I., (2014), "Consistency Analysis and Improvement of Vision-aided Inertial Navigation", *IEEE Transactions on Robotics*, 30(1), 158–176.conference Name: IEEE Transactions on Robotics.
- [213] Liu Z., Zhou Q., Qin Y., El.Sheimy N., (2017), "Vision-aided inertial navigation system with point and vertical line observations for land vehicle applications", *China Satellite Navigation Conference (CSNC) 2017 Proceedings: Volume II*, 445–457, Singapore, May.
- [214] Babu B.P.W., Cyganski D., Duckworth J., Kim S., (2018), "Detection and Resolution of Motion Conflict in Visual Inertial Odometry", 2018 IEEE International Conference on Robotics and Automation (ICRA), 996–1002, Brisbane, Australia, May.

- [215] Konda K.R., Memisevic R., (2015), "Learning visual odometry with a convolutional network", International Conference on Computer Vision Theory and Applications, 486-490, Berlin, Germany, May.
- [216] Kendall A., Grimes M., Cipolla R., (2016), "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization", arXiv:1505.07427 [cs], .arXiv: 1505.07427.
- [217] Costante G., Mancini M., Valigi P., Ciarfuglia T.A., (2016), "Exploring Representation Learning With CNNs for Frame-to-Frame Ego-Motion Estimation", IEEE Robotics and Automation Letters, 1(1), 18–25.
- [218] Clark R., Wang S., Wen H., Markham A., Trigoni N., (2017), "VINet: Visual-Inertial Odometry as a Sequence-to-Sequence Learning Problem", arXiv:1701.08376 [cs], .arXiv: 1701.08376.
- [219] Zelenka R., (1992), "Integration of radar altimeter, precision navigation, and digital terrain data for low-altitude flight", 1st, .
- [220] Oh, Sung, Lee, Lee, Lee, Yu, (2019), "Accurate Measurement Calculation Method for Interferometric Radar Altimeter-Based Terrain Referenced Navigation", Sensors, 19(7), 1688.
- [221] Turan B., Kutay A.T., (2016), "Particle filter studies on terrain referenced navigation", 2016 IEEE/ION Position, Location and Navigation Symposium (PLANS), 949-954, Savannah, GA, USA, August.
- [222] Yoo Y.M., Park C.G., (2015), "Improvement of terrain referenced navigation using a Point Mass Filter with grid adaptation", International Journal of Control, Automation and Systems, 13(5), 1173–1181.
- [223] Jeon H.C., Park W.J., Park C.G., (2018), "Grid design for efficient and accurate point mass filter-based terrain referenced navigation", IEEE Sensors Journal, 18(4), 1731–1738.
- [224] Teixeira F.C., Quintas J., Maurya P., Pascoal A., (2017), "Robust particle filter formulations with application to terrain-aided navigation", International Journal of Adaptive Control and Signal Processing, 31(4), 608–651. _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/acs.2692>.
- [225] Zhao L., Gao N., Huang B., Wang Q., Zhou J., (2015), "A novel terrain-aided navigation algorithm combined with the tercom algorithm and particle filter", IEEE Sensors Journal, 15(2), 1124–1131.
- [226] Zhao L., Gao N., Huang B., Wang Q., Zhou J., (2015), "A Novel Terrain-Aided Navigation Algorithm Combined With the TERCOM Algorithm and Particle Filter", IEEE Sensors Journal, 15(2), 1124–1131. conference Name: IEEE Sensors Journal.
- [227] Yu P.J., Chen Z., Hung J.C., (1991), "Performance evaluation of six terrain stochastic linearization techniques for tan", Proceedings of the IEEE 1991 National Aerospace and Electronics Conference NAECON 1991, 382-388 vol.1, Dayton, OH, USA, May.
- [228] Tian X., Bar.Shalom Y., (2010), "On algorithms for asynchronous Track-to-Track Fusion", 2010 13th International Conference on Information Fusion, 1–8, Edinburgh, UK, Jul..

BIOGRAPHY

Hasan Hüseyin Sönmez received the B.Sc. degree from Gebze Institute of Technology, Kocaeli, Türkiye, in 2013 and the M.Sc. degree from Gebze Technical University, Kocaeli, Türkiye, in 2015, both in electronics engineering. He is currently working toward the Ph.D. degree in electronics engineering in institute of natural and applied science at Gebze Technical University (GTU), Kocaeli, Turkey. He is currently employed as a navigation and guidance engineer at Aselsan Inc., Ankara, Türkiye. His research interests include target tracking, data fusion, Bayesian filtering, sensor control.

APPENDICES

Appendix A: Journal Articles (SCI/SCI-EXP) Based on the Thesis

The journal articles are as follows:

1. Sönmez, H. H., Hocaoglu, A. K., (2023), "Hölder divergence-based reward function for Poisson RFSs and application to multitarget sensor management", IEEE Sensors Journal, 23(9), pp. 9999-10008.
2. Sönmez, H. H., Hocaoglu, A. K., (2022), "Asynchronous track-to-track association algorithm based on reference topology feature", SIViP, 16(3), 789–796.
3. Sönmez, H. H., Hocaoglu, A. K., (2020), "Analysis of performance criteria for optimization based bearing only target tracking algorithms", arXiv preprint arXiv:2002.09183.
4. Sönmez, H. H., Sönmez, T., and Hocaoglu, A. K., (2023), "Distributed heterogeneous sensor fusion for assured navigation applications", To be submitted to IET Radar, Sonar & Navigation, 1-15.
5. Sönmez, H. H., Hocaoglu, A. K., (2023), "Multiple-cluster target tracking based on random finite set formulation and Dirichlet process mixtures", To be submitted to Information Fusion, 1-15.

Appendix B: Conference Proceedings Based on/related to the Thesis

The international conference proceedings based on the thesis are as follows:

1. Gültekin, A., Büyüktepe, İ. C., Sönmez, H. H., and Hocaoglu, A. K., (2018), "Vehicle tracking on video sequences via subspace learning," 2018 26th Signal Processing and Communications Applications Conference (SIU), pp. 1-4, İzmir, Türkiye, 02-05 May. doi: 10.1109/SIU.2018.8404205.
2. Sönmez, H. H., Turgut, K., and Hocaoglu, A. K., (2017), "Improving track initiation performance by track validation algorithms for multi-target tracking in heavy clutter," 2017 25th Signal Processing and Communications Applications Conference (SIU), pp. 1-4, Antalya, Türkiye, 15-18 May. doi: 10.1109/SIU.2017.7960467.

3. Sönmez, H. H., Hoccoğlu, A. K., and Genç, H. M., (2017), "A new solution method for the passive target motion analysis problem using evolutionary strategies," 2017 25th Signal Processing and Communications Applications Conference (SIU), pp. 1-4, Antalya, Türkiye, 15-18 May. doi: 10.1109/SIU.2017.7960401.

The national conference proceedings based on the thesis are as follows:

4. Sönmez, H. H., Sönmez, T., (2022), "Visual Aided Navigation Application and Sensitivity Analysis for Aerial Vehicles", SAVTEK 2022, 10th Defense Technologies Congress, pp. 1-14, Ankara, Türkiye, 13-15 September.