DECENTRALIZED RIEMANNIAN PARTICLE FILTERING WITH APPLICATIONS TO MULTI-AGENT LOCALIZATION

DISSERTATION

Martin J. Eilders, Civilian, USAF

AFIT/DEE/ENG/12-05

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT/DEE/ENG/12-05

DECENTRALIZED RIEMANNIAN PARTICLE FILTERING WITH APPLICATIONS TO MULTI-AGENT LOCALIZATION

DISSERTATION

Presented to the Faculty
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
In Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy

Martin J. Eilders, BSEE, MSEE
Civilian, USAF

June 2012

AFIT/DEE/ENG/12-05

Decentralized Riemannian Particle Filtering with Applications to Multi-Agent Localization

Martin J. Eilders, BSEE, MSEE
Civilian, USAF

Approved:

John F. Raquet
John F. Raquet, Ph.D. (Chairman)

30 MAY 2012
Date

Richard K. Martin
Richard K. Martin, Ph.D. (Member)

23 May 2012
Date

Mark E. Oxley
Mark E. Oxley, Ph.D. (Member)

23 May 2012
Date

Accepted:

MU Thomas
M. U. Thomas
Dean, Graduate School of Engineering
and Management

4 Jun 2012
Date

AFIT/DEE/ENG/12-05

Abstract

The primary focus of this research is to develop consistent nonlinear decentralized particle filtering approaches to the problem of multiple agent localization. A key aspect in our development is the use of Riemannian geometry to exploit the inherently non-Euclidean characteristics that are typical when considering multiple agent localization scenarios. A decentralized formulation is considered due to the practical advantages it provides over centralized fusion architectures.

Inspiration is taken from the relatively new field of information geometry and the more established research field of computer vision. Differential geometric tools such as manifolds, geodesics, tangent spaces, exponential, and logarithmic mappings are used extensively to describe probabilistic quantities. Numerous probabilistic parameterizations were identified, settling on the efficient square-root probability density function parameterization. The square-root parameterization has the benefit of allowing filter calculations to be carried out on the well-studied Riemannian unit hypersphere. A key advantage for selecting the unit hypersphere is that it permits closed-form calculations, a characteristic that is not shared by current solution approaches.

Through the use of the Riemannian geometry of the unit hypersphere, we are able to demonstrate the ability to produce estimates that are not overly optimistic. Results are presented that clearly show the ability of the proposed approaches to outperform current state-of-the-art decentralized particle filtering methods. In particular, results are presented that emphasize the achievable improvement in estimation error, estimator consistency, and required computational burden.

Acknowledgements

No man is an island. Truer words have never been spoken in the context of this research endeavor. The mere thought of attempting to pursue a doctoral degree would not have been realized if it had not been for the help and encouragement that I received along the way. I would like to acknowledge the following individuals in particular.

First, I would like to thank the Air Force Research Laboratory Munitions Directorate for providing me this opportunity. For an employer to provide the ability to go off to school for 2 years uninterrupted is almost unheard of. This was truly an unique opportunity, one that was deeply appreciated by both my family and I.

The willingness of my research advisor, Dr. John Raquet, to work with me under unusual circumstances, on topics that were initially well outside of my technical comfort zone has been greatly appreciated. His weekly phone conversations served to motivate me and to remind me to keep my eye on the prize.

Finally, in no way would this have been remotely possible without the continuous and unconditional love and support of my family. To my wife and daughters, words can not express my appreciation for all that you have done and tolerated. This dissertation is as much your accomplishment as it is mine.

Martin J. Eilders

Table of Contents

Page

Abstract iv

Acknowledgements v

List of Figures x

List of Tables xviii

I. Introduction 1
- 1.1 Chapter Overview 1
- 1.2 Decentralized Fusion 3
- 1.3 Proposed Approach 5
- 1.4 Research Contributions 5
- 1.5 Dissertation Outline 6

II. Background 8
- 2.1 Chapter Overview 8
- 2.2 General Differential Geometry 8
  - 2.2.1 Manifolds 8
  - 2.2.2 Tangent Spaces 12
  - 2.2.3 Connections 13
  - 2.2.4 Geodesics 14
  - 2.2.5 Manifold Mapping Operators 15
  - 2.2.6 Completeness Assumption 16
  - 2.2.7 Comparison of Geometries 17
- 2.3 The Unit Sphere: $\mathcal{S}^n$ 18
  - 2.3.1 Definitions & Relationships 18
  - 2.3.2 Analytical Tools 20
- 2.4 Data Fusion 26
  - 2.4.1 Common Models for Data Fusion 26
  - 2.4.2 Non-Bayesian Data Fusion Methods 29
  - 2.4.3 Parametric Bayesian Data Fusion Methods 29
- 2.5 Particle Filters 32
  - 2.5.1 Monte Carlo Integration 34
  - 2.5.2 Importance Sampling 37
  - 2.5.3 Sequential Importance Sampling (SIS) 40

Page

2.5.4 Resampling 44
2.5.5 Sample Importance Resampling (SIR) 47
2.5.6 Converting Particles to Probabilities 49
2.5.7 Discussion 54
2.6 Decentralized Particle Filtering 55
2.6.1 Key Components of Fusion Equation 56
2.6.2 Interpretation of Fusion Equation 59
2.7 Information Measures 60
2.7.1 Kullback-Leibler Divergence 61
2.7.2 Hellinger Distance 61
2.7.3 Bhattacharyya Divergence 62
2.7.4 Chernoff Divergence 63
2.8 Detailed Literature Survey of Closely Related Efforts 63
2.8.1 What is a Consistent Estimate? 64
2.8.2 Conservative Data Fusion Methods 64
2.8.3 Nonlinear Filtering and Differential Geometry . . 71
2.8.4 Particle Filtering and Decentralized Data Fusion . 75
2.9 Summary 80

III. Novel Approaches to Decentralized Particle Filtering 82
3.1 Chapter Overview 82
3.2 The Geometry of Existing Methods 82
3.3 Riemannian Structure of Probability Spaces 84
3.4 Intrinsic Statistics on Manifolds 86
3.5 Decentralized Riemannian Particle Filter 89
3.5.1 Global Update Module 90
3.5.2 Conceptual Preview 91
3.5.3 From Particles to Probabilities 92
3.6 Algorithm I: Intrinsic Data Fusion Approach 96
3.6.1 Projecting Onto $\mathcal{S}^n$ 96
3.6.2 Tangent Vector Calculation 97
3.6.3 Mean Tangent Vector Calculation 98
3.6.4 Projection Back Onto $\mathcal{S}^n$ 98
3.6.5 Calculate Error 98
3.6.6 Deciding to Continue or Not 99
3.6.7 Returning Back to Where We Started 99
3.6.8 A Brief Discussion 100
3.7 Algorithm II: Information Geometric Approach 101

Page

3.7.1 Establishing Information Relationships 101
3.7.2 Conceptual Motivation 103
3.7.3 Performing Geodesic Interpolation 104
3.7.4 Information Distance Concepts for Fusion 106
3.7.5 Fusing ψ's for Density Selection 107
3.8 Updating Particle Weights 112
3.8.1 Weight Update With MAP Estimation 112
3.8.2 Weight Update With Geodesic Interpolation . . . 113
3.8.3 Particle Resampling 114
3.9 Summary 115

IV. Detailed Simulation Analysis and Discussion of Results 117
4.1 Chapter Overview 117
4.2 Simulation Scenarios, Models, and Parameters 117
4.2.1 Simulation Scenario 117
4.2.2 Process Model 119
4.2.3 Measurement Model 120
4.3 Setting the Stage: Parameters & Assumptions 121
4.4 Comparing Gradient and Information Based Algorithms . . 124
4.4.1 Direct Comparison 124
4.4.2 A Closer Look 127
4.5 Algorithm Operational Integrity 129
4.5.1 Ability to Perform Consistent Estimation 129
4.5.2 Individual vs. Centralized vs. Decentralized . . . 131
4.5.3 Impact of Number of Particles 133
4.5.4 Impact of Number of Agents 137
4.5.5 Information Analysis 137
4.6 Performance Under Various Measurement Scenarios 142
4.6.1 Range and Bearing Case 142
4.6.2 Range Only Case 144
4.6.3 Bearing Only Case 146
4.7 State-of-the-Art Comparison 152
4.8 Summary 156

V. Conclusions 157
5.1 Introduction 157
5.2 Value Added by the Research Effort 158
5.3 Areas Worthy of Future Considerations 160
5.4 Final Thoughts 161

Page

Appendix A. Topics From Topology and Real Analysis 163
A.1 Introduction 163
A.2 Definitions From Analysis 163
A.3 Definitions From Topology 166
A.4 Norms, Metrics, and Inner Products 166
A.5 Hilbert and Banach Spaces 168

Bibliography 169

List of Figures

Figure Page

2.1. Depiction of a n dimensional manifold $\mathcal{S}^n$with two neighborhoods $\mathcal{U}_1$ and $\mathcal{U}_2$ along with mappings ϕ_1 and ϕ_2 to elements in $\mathbb{R}^n$. . . . 10

2.2. A representation of a manifold $\mathcal{S}$ with point $\mathbf{p}$. At $\mathbf{p}$ is a tangent space $\mathcal{T}_{\mathbf{p}}(\mathcal{S})$ that is comprised of all vectors that are tangent to $\mathcal{S}$ at point $\mathbf{p}$ and are denoted by $\boldsymbol{\tau}_1, \boldsymbol{\tau}_2$, and $\boldsymbol{\tau}_3$. (inspired by references [12], [155]) 12

2.3. The connection that establishes a relationship between the two tangent spaces. Note that all of the tangent vectors $\dot{\gamma}(t)$ have the same magnitude and are parallel to one another. This observation is discussed in the next section. 14

2.4. A geodesic γ is defined as the shortest path between points $\mathbf{p}$ and $\mathbf{p}'$ on $\mathcal{S}$. 15

2.5. An exponential mapping of the tangent vector $\boldsymbol{\tau}_1$ with origin at point ψ_1 in tangent space $\mathcal{T}_{\mathbf{p}}(\mathcal{S})$. Additionally, it shows the logarithmic mapping of the corresponding point on the manifold $\mathcal{S}$ to the end point of tangent vector $\boldsymbol{\tau}_1$. 16

2.6. The unit sphere embedded in $\mathbb{R}^3$. By definition, all principle axis are unit length. 20

2.7. The tangent space $\mathcal{T}_{\mathbf{p}}(\mathcal{S}^n)$ at point $\mathbf{p}$ on the unit sphere, $\mathbf{p} \in \mathcal{S}^n$, embedded in $\mathbb{R}^{n+1}$. 21

2.8. The geodesic $\gamma(t)$ connecting points $\mathbf{p}$ and $\mathbf{p}'$ on the unit sphere $\mathcal{S}^n$. 23

2.9. The exponential map and logarithmic map on the unit sphere $\mathcal{S}^n$. . 24

2.10. Data fusion model published by the Joint Directors of Laboratories (JDL) [294]. 27

2.11. Revised data fusion model published by the JDL in 1998, and can be found in [266]. 28

2.12. Probability density, 1000 Monte Carlo samples, and contour plot of a Gaussian pdf. 36

Figure Page

2.13. Importance sampling with 100 samples. 40

2.14. Importance sampling with 1000 samples. 41

2.15. Importance sampling with 10000 samples. 41

2.16. Illustration of the phenomenon known as particle degeneracy. . . . 45

2.17. Demonstration of the resampling process used in particle filtering algorithms. 47

2.18. A Gaussian mixture model implementation with the Expectation Maximization algorithm. The scenario considered utilized 5000 samples. 52

2.19. Kernel density estimation with Gaussian kernels. The scenario considered utilized 10 kernels with all having unit variance and mean values given by $[-4.5, -3.1, -2.3, -0.4, 0.5, 1.9, 3.4, 5.1, 6.2, 8.6]$. 54

2.20. Figure representing the result of multiplying two collections of particles that are not collocated. 57

2.21. Conditional probability densities typical of the dependent data fusion process. 58

3.1. A block diagram of the typical processes associated with a sensing agent in an ad-hoc communication topology. 89

3.2. Block diagram of the algorithmic steps required during global data fusion. 91

3.3. Figure representing the conceptual process of global data fusion. . 93

3.4. Typical histogram plot of a landmark state produced by the simulation. 94

3.5. A block diagram of the necessary steps for performing global fusion. 96

3.6. A block diagram of the necessary steps for updating particles after the fused density has been selected. 99

3.7. Two probability density functions used in the example for demonstrating the construction of a geodesic. 105

3.8. Every tenth sample along the geodesic constructed between two initial probability density functions in the example. 106

Figure Page

3.9. Two 1D Gaussian pdf's used in the creation of Figure 3.10. 110

3.10. A graphical depiction of the differential geometric relationships between various information divergence functions. The blue curve represents the geodesic that connects probability density functions $\mathbf{p}$ and $\mathbf{q}$ and parameterized by ω. This figure is a recreation of a figure that can be found in the works of Sinanovic and Johnson [134]. 111

3.11. A block diagram representing the algorithmic flow of a decentralized particle filtering algorithm based on information measures and geodesic interpolation. 112

4.1. A typical simulation scenario. 118

4.2. Indication of random time epoch of when the agents were allowed to communicate with one another. 123

4.3. A Single Run example showing agent1's vehicle specific state estimation error along with the $\pm 1\sigma$ bounds for the information-based algorithm presented in Section 3.6. 125

4.4. A Single Run example showing agent1's landmark specific state estimation error (red) along with the $\pm 1\sigma$ bounds (black) for the information-based algorithm presented in Section 3.6. 126

4.5. All 100 Monte Carlo trials showing agent1's x-position state estimation errors (blue), mean filter generated $\pm 1\sigma$ bounds (red), the ensemble error $\pm 1\sigma$ bounds (yellow), and the mean estimation error (green) for the information-based algorithm presented in Section 3.6. 126

4.6. Comparison of Agent1's x-position state uncertainty obtained by the gradient-based, information based, and naïve algorithms presented in Sections 3.6 and 3.7 respectively in meters. 127

4.7. Comparison of Agent1's y-position state uncertainty obtained by the gradient-based, information based, and naïve algorithms presented in Sections 3.6 and 3.7 respectively in meters. 128

4.8. Comparison of Agent1's heading state uncertainty obtained by the gradient-based, information based, and naïve based algorithms presented in Sections 3.6 and 3.7 respectively in degrees. 128

Figure Page

4.9. A single dimensional probability density for the x-position of landmark1 obtained from the gradient-based algorithm where the green line represents the true position and the red line represents the corresponding estimated position. 129

4.10. A single dimensional probability density for the x-position of landmark1 obtained from the information-based algorithm where the green line represents the true position and the red line represents the corresponding estimated position. 130

4.11. Agent1 consistency test for x-position state uncertainty as the ratio between the uncertainties obtained when using a centralized and decentralized processing architecture. 131

4.12. Agent1 consistency test for y-position state uncertainty as the ratio between the uncertainties obtained when using a centralized and decentralized processing architecture. 132

4.13. Agent1 consistency test for heading angle state uncertainty as the ratio between the uncertainties obtained when using a centralized and decentralized processing architecture. 132

4.14. Comparison between agent1 x-position uncertainties obtained by implementing a centralized processing, a decentralized processing, and an individual processing scenario. 133

4.15. Comparison between agent1 y-position uncertainties obtained by implementing a centralized processing, a decentralized processing, and an individual processing scenario. 134

4.16. Comparison between agent1 heading angle uncertainties obtained by implementing a centralized processing, a decentralized processing, and an individual processing scenario. 134

4.17. Results and analysis of the impact that the number of particles has on the estimation accuracy for agent1's X position. 136

4.18. Results and analysis of the impact on the estimation accuracy versus the number of local neighborhood agents considered during a particular global update cycle for agent1's Y position. 136

4.19. Results and analysis of the impact on the estimation accuracy versus the number of local neighborhood agents considered during a particular global update cycle for agent1's heading. 137

4.20. Results and analysis of the impact on the estimation accuracy versus the number of local neighborhood agents considered during a particular global update cycle for agent1's X position. 138

4.21. Results and analysis of the impact on the estimation accuracy versus the number of local neighborhood agents considered during a particular global update cycle for agent1's Y position. 138

4.22. Results and analysis of the impact on the estimation accuracy versus the number of local neighborhood agents considered during a particular global update cycle for agent1's heading. 139

4.23. Results and analysis of the impact on the estimation accuracy versus the number of local neighborhood agents considered for agent1's X position in the form of a bar plot. 139

4.24. Results and analysis of the impact on the estimation accuracy versus the number of local neighborhood agents considered for agent1's Y position in the form of a bar plot. 140

4.25. Results and analysis of the impact on the estimation accuracy versus the number of local neighborhood agents considered for agent1's heading in the form of a bar plot. 140

4.26. A comparison of the Shannon entropy associated with the centralized, decentralized, and no communication processing scenarios. . 141

4.27. A comparison of the Kullback-Leibler divergence for the decentralized and no communication processing scenarios, and for both cases the centralized case was used as the reference or target case. . 142

4.28. Agent1 x-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation (red), and ensemble mean estimation error (blue) for the range and bearing measurement case. 143

4.29. Agent1 y-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation (red), and ensemble mean estimation error (blue) for the range and bearing measurement case. 143

4.30. Agent1 heading angle state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation (red), and ensemble mean estimation error (blue) for the range and bearing measurement case. 144

4.31. Agent1's landmark1 x-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation (red), and ensemble mean estimation error (blue) for the range and bearing measurement case. 145

4.32. Agent1's landmark1 y-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation (red), and ensemble mean estimation error (blue) for the range and bearing measurement case. 145

4.33. Agent1 x-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation (red), and ensemble mean estimation error (blue) for the range only measurement case. 146

4.34. Agent1 y-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation (red), and ensemble mean estimation error (blue) for the range only measurement case. 147

4.35. Agent1 heading angle state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation (red), and ensemble mean estimation error (blue) for the range only measurement case. 147

4.36. Agent1's landmark1 x-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation (red), and ensemble mean estimation error (blue) for the range only measurement case. 148

Figure Page

4.37. Agent1's landmark1 y-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation (red), and ensemble mean estimation error (blue) for the range only measurement case. 148

4.38. Agent1 x-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation (red), and ensemble mean estimation error (blue) for the bearings only measurement case. 149

4.39. Agent1 y-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation (red), and ensemble mean estimation error (blue) for the bearings only measurement case. 150

4.40. Agent1 heading angle state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation (red), and ensemble mean estimation error (blue) for the bearings only measurement case. 150

4.41. Agent1's landmark1 x-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation (red), and ensemble mean estimation error (blue) for the bearings only measurement case. 151

4.42. Agent1's landmark1 y-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation (red), and ensemble mean estimation error (blue) for the bearings only measurement case. 151

4.43. A comparison of agent1's x-position mean filter generated 1σ standard deviations over 100 Monte Carlo runs for the proposed information based unit hypersphere algorithm, traditional Covariance Intersection in an EKF framework, and Gaussian Mixture Model Particle Filter (GMMPF). This particular figure presents the x-position filter generated estimation errors for agent1. The units in the figure are meters. 154

Figure Page

4.44. A comparison of agent1's y-position mean filter generated 1σ standard deviations over 100 Monte Carlo runs for the proposed information based unit hypersphere algorithm, traditional Covariance Intersection in an EKF framework, and Gaussian Mixture Model Particle Filter (GMMPF). This particular figure presents the x-position filter generated estimation errors for agent1. The units in the figure are meters. 154

4.45. A comparison of the mean filter generated 1σ standard deviations over 100 Monte Carlo runs for the proposed information based unit hypersphere algorithm, traditional Covariance Intersection in an EKF framework, and Gaussian Mixture Model Particle Filter (GMMPF). This particular figure presents the x-position filter generated estimation errors for agent1. The units in the figure are meters. 155

A.1. An injective mapping. The original figure can be found in reference [202]. 164

A.2. A surjective mapping. The original figure can be found in reference [202]. 165

List of Tables

Table Page

2.1. Relationship between arithmetic operations of addition and subtraction that are available in Euclidean vector spaces and the corresponding operations that are available on Riemannian manifolds. (After [299]) 17

2.2. Key tools that are available for working on the unit n-Sphere. This table is a partial replication of a table found in [145]. 25

2.3. Initial weights for a collection of 15 particles and the resulting particles from implementing a resampling step. 46

4.1. Parameters values used in the simulation 122

4.2. PCCI mean run times varying measurements & particles 153

4.3. DRPF mean run times varying measurements & particles 153

Decentralized Riemannian Particle Filtering with Applications to Multi-Agent Localization

I. Introduction

1.1 Chapter Overview

Technologically speaking, we are in the midst of some truly exciting times. The ability to harvest, process, store, and disseminate data is unparalleled from any other time in history. The magnitude of the most recent technology growth-spurt has resulted in the streamlining of multiple enabling technologies. The continual technological advances in both hardware and software has made the realization of multi-agent systems in complex environments more realistic than in previous years. From a Department of Defense (DoD) perspective, multi-agent systems offer a potential launching point for the various Network Enabling Capabilities (NEC), that have been deemed necessary for realizing the Global Information Grid (GIG).

Net-centric warfare (NCW) doctrine levies considerable technological challenges on the way data is shared, processed, and stored. The sheer volume of data sources available requires that data processing functions be implemented on lower tier components. The reallocation of data processing tasks to various individual components while still keeping decision-makers informed, has proven to be challenging. Numerous technical challenges can be exemplified through applications involving multiple agent systems. The manner in which multi-agent systems interact, communicate, and their levels of autonomy are all questions receiving active research attention. Two common threads among all of these research topics is system architectures and data fusion methodologies.

Multisensor data fusion (MSDF) is concerned with assimilating data from multiple sensors/sources in order to obtain a consistent and coherent environmental representation.

Prevalent throughout many military systems, MSDF methods are utilized in a series of tasks to include battlefield surveillance missions [39], multi-target tracking (MTT) [271], automatic target recognition (ATR) [116], and navigation of manned and unmanned systems [172], just to name a few. Clearly the list of applications requiring some form of data fusion process is rather extensive. The use for data fusion processes has proven valuable in many traditionally disjoint scientific and engineering disciplines. Nevertheless, the wide spread use of data fusion processes has contributed to the numerous algorithm alternatives, as well as to the current inability to produce standardizations of vocabulary, solution approaches, and Bayesian filtering models.

A challenging problem in its own right, now add to the complexity of MSDF the arduous task of distributing available data sources across multiple mobile sensing platforms. In the context of military applications, multi-agent systems present formidable challenges. For example, agents intended for battlefield operations will almost surely be required to be cheap and disposable, virtually guaranteeing that resources like computational power, data storage facilities, and power resources will be limited. Scarcity of resources means methods to manage them efficiently will need to be developed. Cheap agents can be used in large numbers, hence a system architecture that can scale to large agent populations will need to be available. Environments in which multi-agent systems will operate can be expected to be complex and hostile, and coupled with the mobility of agents will require a dynamic communications topology to effectively operate.

The focus of this dissertation is on a subset of the larger multiple agent data fusion paradigm. To be more precise, how to implement consistent decentralized particle filtering algorithms when an arbitrary number of agents are allowed to stochastically communicate data with one another is the principal focus of this research. Furthermore, the individual agents will need to fuse the data resulting from other agents with the data obtained from their own sensor suite.

1.2 Decentralized Fusion

Available data fusion architectures take on several different manifestations in the literature. Perhaps the simplest system architecture is one where all of the necessary blending of available data occurs at a common processing station, from which results are disseminated to external customers [217]. This type of processing scheme is known as centralized, since there exists a common processing facility privileged to the most complete view of the environment [203]. Access to all data sources gives the central processing core control over the entire system decision-making process [197]. A centralized architecture may be adequate for some applications. However, with increasing processing intricacies and reduction in hardware costs, the argument can be made that alternative architectures may be more appropriate [172]. Another argument for considering alternative architectures is that a centralized architecture contains a single point of failure which, from a military perspective, is both tactically and strategically risky [293]. As a result, centralized architectures are becoming less of a preferred option in modern multi-agent networks [203].

Decentralized data fusion (DDF) networks are more reliable than centralized networks and can operate successfully under conditions that would render centralized networks useless [89]. The primary source of robustness stems from the removal of any single point of failure [112]. Removing the need for a single processing agent responsible for the entire network permits the loss of an arbitrary subset of data sources, without incapacitating the entire network [91]. Second, DDF networks distribute the burden of operating functions across the network [288]. The immediate benefit is to the lessening of bandwidth constraints in arbitrary subsets of agents [205]. Third, the pure nature of DDF networks implies modularity [186]. Since knowledge of a local subset of the entire network is all that is required of any one data source, the network can become globally dynamic without having an impact locally (*i.e.,* modular) [182]. The added benefit of modularity is that it inherently permits the network to be made up of radically different data sources without having to consider their differences explicitly [90]. Once an architecture has been identified, one will need to determine an appropriate fusion mechanism.

Decentralized filtering has recently incorporated sample based-techniques such as particle filters [276], [115], [254], [107], with varying degrees of success [297], [115]. Particle filters are a nonlinear filtering method based on a Bayesian probability formulation. Particle filters pose additional research questions over more common “parametric methods". The questions of how to represent and relay the information content in a collection of particles has provoked sporadic research attention and certainly is deserving of more.

The central idea behind particle filtering is to approximate probability densities with a set of independent and identically distributed (i.i.d.) random samples known as particles. Under the Bayesian formulation, the particles are used to propagate and update filtering densities. Their ability to represent arbitrary probability densities to any desired degree of accuracy and ease of implementation have contributed to their increased use in multi-agent decentralized data fusion scenarios [43], [120], [168].

The most notable disadvantage of using particle filters is the computational burden they impose by requiring large numbers of samples to represent probability densities of even modest dimensions. However, the availability and affordability of powerful computing resources is making questions concerning their computational burden less important, and making them ideal candidates for use in multi-agent DDF formulations. Particle filters will be discussed in greater detail in Section 2.5.

Many practical situations will often not be governed by linear models nor will system disturbances be accurately represented with Gaussian statistics. Furthermore, the types of probability density function that are surely to be encountered will likely be characterized as exotic multi-modal densities requiring novel solution methods. The surplus of nontraditional representations still requires innovative solution approaches in order to allow practical decentralized particle filtering implementations.

1.3 Proposed Approach

Motivating this dissertation is the lack of available practical decentralized particle filtering techniques for use in multi-agent systems. The number of available algorithms is limited when compared to the availability of their parametric counterparts. The filtering benefits that particle filters offer along with the ability to realize them in multi-agent scenarios are why they are chosen as the fusion method of choice in this dissertation.

There is an inherent geometrical structure associated with general nonlinear filtering techniques, and with particle filtering algorithms specifically. This geometry is exploited to increase levels of efficiency and robustness, through the use of differential geometry. In particular, the wealth of analysis tools made available through the use of Riemannian geometry are utilized.

The use of the synergy that exists between differential geometry and particle filtering techniques is accomplished through projection operations. In the proposed approach, data fusion doesn't occur in traditional state space; instead, it occurs by projecting filtering densities onto alternative fusion surfaces. The primary surface used is the n-dimensional unit hypersphere. Once on the surface of the sphere, differential geometric tools and information theoretics are used to describe relationships between probability densities, and are ultimately used to select filtering densities for the fusion process.

1.4 Research Contributions

The following is a list of proposed novel research contributions of the research presented in this dissertation. To this author's knowledge, none of the methods motivating the proposed contributions have been realized and/or published within the available technical literature.

1. We are able to demonstrate a never before used general framework for performing particle filtering in decentralized architectures that is based on a non-Euclidean geometric interpretation of decentralized data fusion. Current decentralized filtering

methods represent a dichotomy of techniques. The first class of methods requires the ability to linearize models so that Kalman based methods can be used for decentralized data fusion. The second class of methods makes use of particle filtering technology by requiring that complex filtering densities be represented with mixture models for decentralized data fusion. Our framework relies on no such requirements.

2. Through our choice of Riemannian interpretation of the decentralized particle filtering paradigm, we present a new method for conducting particle filtering in decentralized architectures that provides closed form calculations that are currently unavailable in the general case.
3. We were able to adapt an algorithm capable of providing existence and uniqueness guarantees for solutions to the decentralized particle filtering problem under mild assumptions. Existence and uniqueness guarantees are not associated with existing approaches unless under restrictive assumptions to the network topology or available probabilistic representations.
4. We established a technology bridge between multiple research communities with the differential geometric framework that permits access to previously unavailable analysis tools.
5. We demonstrate, through empirical evidence, that an order of magnitude improvement in computational performance over existing approaches is possible with the proposed approach.

1.5 Dissertation Outline

Chapter 2 provides a thorough review of the relevant literature, covering topics such as differential geometry, data fusion, and information theory. Also part of Chapter 2 is a comprehensive literature survey of existing work that is most related to our own.

Chapter 3 provides a systematic development of the proposed novel geometric particle filtering framework and algorithms to be used as a general solution approach to decentralized particle filtering.

Chapter 4 consists of a detailed description of the simulation environment used to validate our approach. Additionally, simulation results and subsequent analysis are also provided. A realistic scenario involving 2D localization of two mobile agents is used to exercise the utility of the geometric particle filtering algorithm. Various operating conditions were chosen, and a discussion of results obtained is given.

In Chapter 5, conclusions are stated, avenues worthy of further research are mentioned with justification, and the contributions of the research are restated. Following the final chapter is an appendix where useful mathematical definitions from topology and analysis are given to assist the reader who is unfamiliar with these topics.

II. Background

2.1 Chapter Overview

In the first chapter, we discussed the technology trend towards networked systems in the framework of multiple data agents. The data fusion problem in a multi-agent system was discussed along with system architectures. Statements regarding the benefits and challenges associated with a decentralized data fusion architecture were made.

This chapter presents the necessary background for understanding differential geometry, and the unit hypersphere. The tools from differential geometry for performing nonlinear filtering are described. The mathematical foundations of nonlinear estimation using Bayesian techniques are developed. A special emphasis is given to particle filtering techniques. Alternative approaches are also mentioned. Methods for sub-optimal decentralized data fusion are further developed with details outlining their utility within multi-agent networks. Furthermore, relevant concepts from information theory are presented in the context of their utility in future filtering presentations. The chapter ends with a chronological presentation of the literature that is most closely related to the proposed work.

2.2 General Differential Geometry

The purpose of this Section is to introduce basic concepts from differential geometry. Topics including manifolds, tangent spaces, geodesics, geodesic distance, exponential maps, logarithmic maps, and others are presented. For more details the curious reader is referred to the introductory works provided by Pressley [230]. More advanced presentations can be found in the texts of William Boothby [42], Manfredo do Carmo [57], and John Lee [166].

2.2.1 Manifolds. A large portion of differential geometry is dedicated to the study of curved surfaces known as manifolds. A manifold has several different definitions depending on what reference one is invoking. However, the differences are often times insignificant and amount to merely vocabulary.

From an intuitive perspective, a manifold is a collection of elements that when examined in a local nature will resemble Euclidean space $\mathbb{R}^n$. The collection of local sets or patches are what are known as charts. The collection of all the charts covering the set of local neighborhoods is known as an atlas. It is the collection of local neighborhoods and the atlas that constitutes a manifold. To make these abstract notions more concrete, the following definitions from Do Carmo [82] and Boothby [42] are presented. The definitions are also represented in Figure 2.1, which contains representations for a manifold $\mathcal{S}$, charts $\mathcal{U}_1$ and $\mathcal{U}_2$, and mappings ϕ_1 and ϕ_2.

Definition 2.2.1 (Chart)**.** *Let $\mathcal{S}$ be a set. A chart of $\mathcal{S}$ is a pair $(\mathcal{U}, \phi)$ where $\mathcal{U} \subset \mathcal{S}$ and ϕ is a bijection between $\mathcal{U}$ and an open set of $\mathbb{R}^n$. $\mathcal{U}$ is the chart's domain and n is the chart's dimension. Given $\mathbf{p} \in \mathcal{U}$, the elements of $\phi(\mathbf{p}) = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)$ are called the coordinates of $\mathbf{p}$ in the chart $(\mathcal{U}, \phi)$.*

Definition 2.2.2 (Compatible Charts)**.** *Two charts $(\mathcal{U}_1, \phi_1)$ and $(\mathcal{U}_2, \phi_2)$ of $\mathcal{S}$, of dimensions n and m, respectively, are smoothly compatible ($\mathcal{C}^\infty$-Compatible) if either $\mathcal{U}_1 \cap \mathcal{U}_2 = \emptyset$ or $\mathcal{U}_1 \cap \mathcal{U}_2 \neq \emptyset$ and:*

1. *$\phi_1(\mathcal{U}_1 \cap \mathcal{U}_2)$ is an open set of $\mathbb{R}^n$,*
2. *$\phi_2(\mathcal{U}_1 \cap \mathcal{U}_2)$ is an open set of $\mathbb{R}^n$,*
3. *$\phi_2 \circ \phi_1^{-1} : \phi_1(\mathcal{U}_1 \cap \mathcal{U}_2) \to \phi_2(\mathcal{U}_1 \cap \mathcal{U}_2)$ is a smooth diffeomorphism (i.e., a C^∞ invertible function with a smooth inverse).*

Definition 2.2.3 (Atlas)**.** *A set $\mathcal{A}$ of pairwise smoothly compatible charts $\{(\mathcal{U}_i, \phi_i), i \in I\}$ such that*

$$\bigcup_{i \in I} \mathcal{U}_i = \mathcal{S} \tag{2.1}$$

is a smooth atlas of $\mathcal{S}$.

Given the above definitions, a more formal definition of a manifold can be stated and can subsequently be found in Boothby [42].

Definition 2.2.4 (Manifold)**.** *A manifold denoted as $\mathcal{S}$ of dimension n, or n-manifold, is a collection of objects known as points. Furthermore, every point resides in an open neighborhood on $\mathcal{S}$ and has a continuous one-to-one mapping to an open set of the reals of dimension n, $\mathbb{R}^n$.*

A general manifold endowed with a well-defined topology is known as a topological manifold, and is defined according to Boothby [42] as

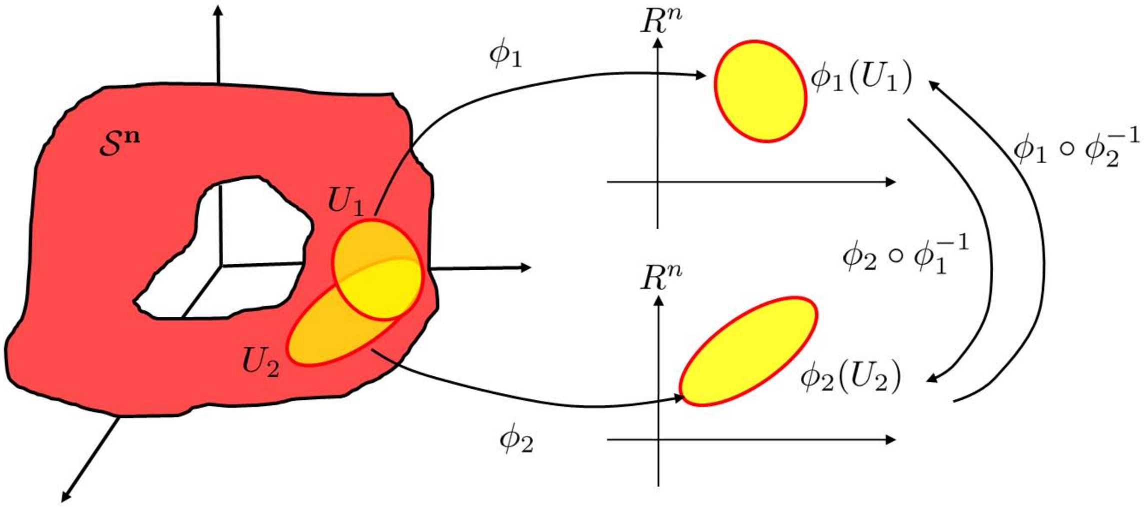

$\mathcal{S}^{\mathbf{n}}$
U_1
U_2
ϕ_1
ϕ_2
R^n
R^n
$\phi_1(U_1)$
$\phi_2(U_2)$
$\phi_1 \circ \phi_2^{-1}$
$\phi_2 \circ \phi_1^{-1}$

Figure 2.1: The n-dimensional manifold $\mathcal{S}^n$, neighborhood charts $\mathcal{U}_1$ and $\mathcal{U}_2$, and homeomorphic mappings ϕ_1 and ϕ_2 (inspired by [12], [155])

Definition 2.2.5 (Topological Manifold)**.** *A manifold $\mathcal{S}^n$ of dimension n, or n-manifold, is said to be a topological manifold if it possesses the following properties:*

1. *$\mathcal{S}^n$ is a Hausdorff space.*
2. *$\mathcal{S}^n$ is locally Euclidean of dimension n.*
3. *$\mathcal{S}^n$ has a countable basis of open sets.*

To help clarify Definition 2.2.5, the following explanations are offered. For a manifold to be a Hausdorff space implies that for any two distinct points there exists two neighborhoods around those points, such that the intersection of the two neighborhoods is the empty set. For the purposes of this dissertation if a manifold has a countable basis of open sets simply, this means that there exists a countable number of coordinate neighborhoods. Referring to Figure 2.1, there are two coordinate neighborhoods shown, and they are defined by the following two pairs $(\mathcal{U}_1, \phi_1)$ and $(\mathcal{U}_2, \phi_2)$. According to the countable basis axiom [82] given in Definition 2.2.5, there can only be a finite number of these coordinate neighborhoods associated with the manifold $\mathcal{S}^n$ shown in Figure 2.1. For the sake of completeness, the axiom also doesn't preclude the possibility of there being a countably infinite number of coordinate neighborhoods. The interested reader can go to references William Boothby [42], Manfredo do Carmo [57], and John Lee [166] if there is a desire to explore the countably infinite aspect further. Before proceeding, the reader is advised that from this point on it is assumed that all manifolds will, at a minimum, adhere to Definition 2.2.5.

A manifold that sufficiently mimics $\mathbb{R}^n$, so that a differential operator can be defined, is known as a differential manifold. More formally, a differential manifold is defined by Morita [255] to be

Definition 2.2.6 (Differentiable Manifold)**.** *Let $\mathcal{S}^n$ be a topological manifold. Furthermore, let $\mathcal{S}^n$ possess an atlas comprised of a collection of charts $\mathcal{A} = \{\phi_i, i \in I\}$ with I denoting the integers and it is called a C^∞ atlas if all of its coordinate changes $\phi_2 \circ \phi_1^{-1}$ are also C^∞ maps or smooth maps. Smooth, in this context, implies the ability to differentiate as many times as desired. It is also stated that the atlas determines a C^∞ structure on $\mathcal{S}$. Hence, a manifold with a C^∞ structure is called a C^∞ differentiable manifold or simply a C^∞ manifold or differential manifold.*

For purposes of this discussion, one can adapt the following hierarchial structure for manifolds. The largest class of manifolds are topological manifolds. Topological manifolds are equipped with just enough structure so that the notion of Euclidean space can be defined, mainly the properties listed previously in Definition 2.2.5. A subset of topological manifolds are differentiable manifolds. Differential manifolds possess additional structure over topological manifolds, as their name suggests. The purpose for imposing the additional structure is usually so that a stronger resemblance to Euclidean space will exists.

Having defined a differentiable manifold $\mathcal{S}^n$, the final class of manifolds to be discussed are Riemannian manifolds. According to Do Carmo [82], Riemannian manifolds are defined according to the following:

Definition 2.2.7 (Riemannian Manifold)**.** *A Riemannian manifold is a differentiable manifold $\mathcal{M}^n$ that has associated to every point $\mathbf{p} \in \mathcal{M}^n$ an inner product $\langle \cdot | \cdot \rangle$ that is symmetric, bilinear, and of positive-definite form (Riemannian metric) on the tangent space $\mathcal{T}_{\mathbf{p}}(\mathcal{M})$. Furthermore, the Riemannian metric for any pair of vector fields VF_1 and VF_2 that are differentiable in a neighborhood $\mathcal{U}$ of point $\mathbf{p} \in \mathcal{M}^n$, the function $\langle \mathrm{VF}_1 | \mathrm{VF}_2 \rangle$ will also be differentiable in the neighborhood $\mathcal{U}$.*

It is common to see the Riemannian metric written as $\langle \boldsymbol{\tau}_1 | \boldsymbol{\tau}_2 \rangle = g_{\mathcal{S}}(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2)$ for tangent vectors $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ in the tangent space associated with point $\mathbf{p}$ (*i.e.*, $(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2) \in \mathcal{T}_{\mathbf{p}}(\mathcal{S})$) [166]. A differentiable manifold, along with a Riemannian metric (if it exists), are together what defines a Riemannian manifold.

In the process of defining a Riemannian manifold, there was a need to utilize tangent spaces and tangent vectors. The concepts for both tangent spaces and tangent vectors are expanded upon in the next section.

2.2.2 Tangent Spaces.

The tangent space, loosely speaking, is a vector space generated through locally linearizing around point $\mathbf{p}$ on manifold $\mathcal{S}$. The tangent space is comprised of all tangent vectors to all curves passing through the point $\mathbf{p}$ at point $\mathbf{p}$. Additionally, the natural basis for the tangent space is formed by the partial derivatives taken with respect to point $\mathbf{p}$, a relationship that is represented by Figure 2.2.

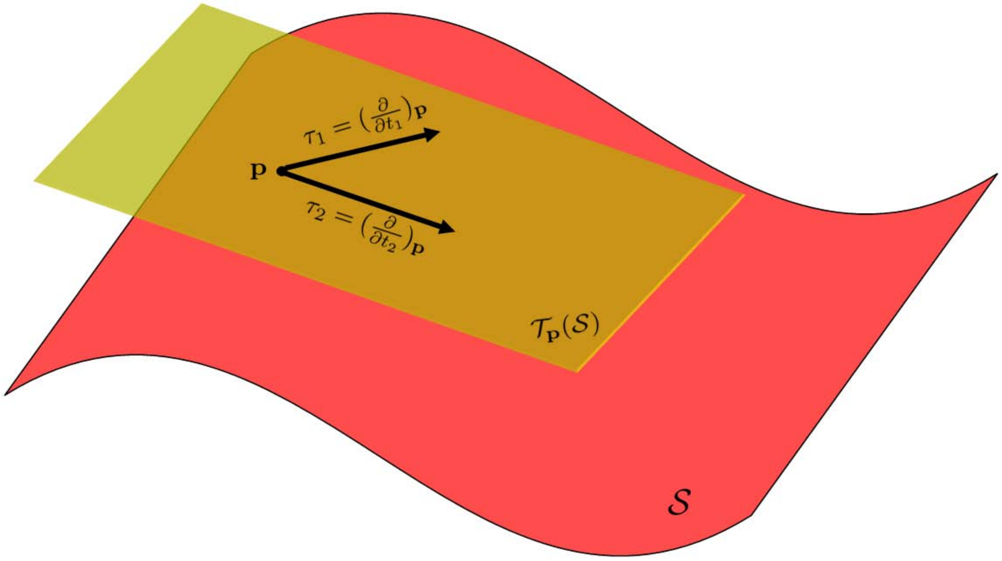

$\tau_1 = (\frac{\partial}{\partial t_1})_{\mathbf{p}}$
$\mathbf{p}$
$\tau_2 = (\frac{\partial}{\partial t_2})_{\mathbf{p}}$
$\mathcal{T}_{\mathbf{p}}(\mathcal{S})$
$\mathcal{S}$

Figure 2.2: Tangent space $\mathcal{T}_{\mathbf{p}}(\mathcal{S})$ for point $\mathbf{p}$ on $\mathcal{S}$ with tangent vectors $\boldsymbol{\tau}_1, \boldsymbol{\tau}_2$, and $\boldsymbol{\tau}_3$

The tangent space is comprised of vectors known as tangent vectors to $\mathcal{S}$ at $\mathbf{p}$ and denoted as $\boldsymbol{\tau}$. A tangent space is formally defined as

Definition 2.2.8 (Tangent Space)**.** *Suppose the following set exists* $\{\mathbf{X_p} : \mathbf{X} \in \mathrm{VF}\}$, *where* $\mathbf{X_p}$ *denotes the vector field associated with* $\mathbf{p} \in \mathcal{S}$, $\mathbf{X}$ *denotes a generic vector field, and* VF *denotes the collection of all vector fields on* $\mathcal{S}$*, and is known as the tangent space to manifold* $\mathcal{S}$ *at point* $\mathbf{p}$ *denoted by:*

$$\mathrm{VF} = \bigcup_{\mathbf{p} \in \mathcal{S}} \mathcal{T}_{\mathbf{p}}(\mathcal{S}). \tag{2.2}$$

In order to fully take advantage of tangent spaces, there needs to be a means of relating elements of two separate tangent spaces to one another. The necessary relationships can be made by use of the operator known as a connection.

2.2.3 Connections. A connection is what allows the discussion of the relationship between two points on $\mathcal{S}$, more specifically a connection provides a mechanism for defining movement from point p to point p' on $\mathcal{S}$. Movement is defined with respect to each point's tangent space. The relationship is defined more formally as [12], [149]:

Definition 2.2.9 (Connection)**.** *On a Riemannian Manifold* $\mathcal{S}$*, a connection (also known as covariant derivative), given a point* $\mathbf{p} \in \mathcal{S}$*, tangent vector* $\boldsymbol{\tau} \in \mathcal{T}_{\mathbf{p}}(\mathcal{S})$*, and a smooth vector field* VF*, is a map* $(\boldsymbol{\tau}, \mathrm{VF}) \mapsto \nabla_{\boldsymbol{\tau}}\mathrm{VF} \in \mathcal{T}_{\mathbf{p}}(\mathcal{S})$*, such that:*

1. $\nabla_{\boldsymbol{\tau}}(\mathrm{VF}_1 + \mathrm{VF}_2) = \nabla_{\boldsymbol{\tau}}\mathrm{VF}_1 + \nabla_{\boldsymbol{\tau}}\mathrm{VF}_2$
2. $\nabla_{(\boldsymbol{\tau}_1+\boldsymbol{\tau}_2)}\mathrm{VF} = \nabla_{\boldsymbol{\tau}_1}\mathrm{VF} + \nabla_{\boldsymbol{\tau}_2}\mathrm{VF}$
3. $\nabla_{\boldsymbol{\tau}}(f\mathrm{VF})(\mathbf{p}) = (\boldsymbol{\tau} f)\mathrm{VF}(\mathbf{p}) + f(\mathbf{p})\nabla_{\boldsymbol{\tau}}\mathrm{VF}$

In other words, given a vector field VF, the connection takes a vector $\boldsymbol{\tau}$ based at point $\mathbf{p}$ to another vector $(\nabla_{\boldsymbol{\tau}}\mathrm{VF})$ based at point $\mathbf{p}$ that depends linearly on the tangent vector $\boldsymbol{\tau}$, linearly on the vector field VF, and follows the Leibnitz rule. Intuitively, a connection is nothing more than a mechanism for transferring the tangent space across $\mathcal{S}$. The connection is what permits discussion of elements of one tangent space with respect to elements of another tangent space, and can be thought of as taking on the same role of the directional derivative of vector field VF in the direction of $\boldsymbol{\tau}$. In fact, the simplest example of a connection is the traditional directional derivative in Euclidean space $\mathbb{R}^n$.

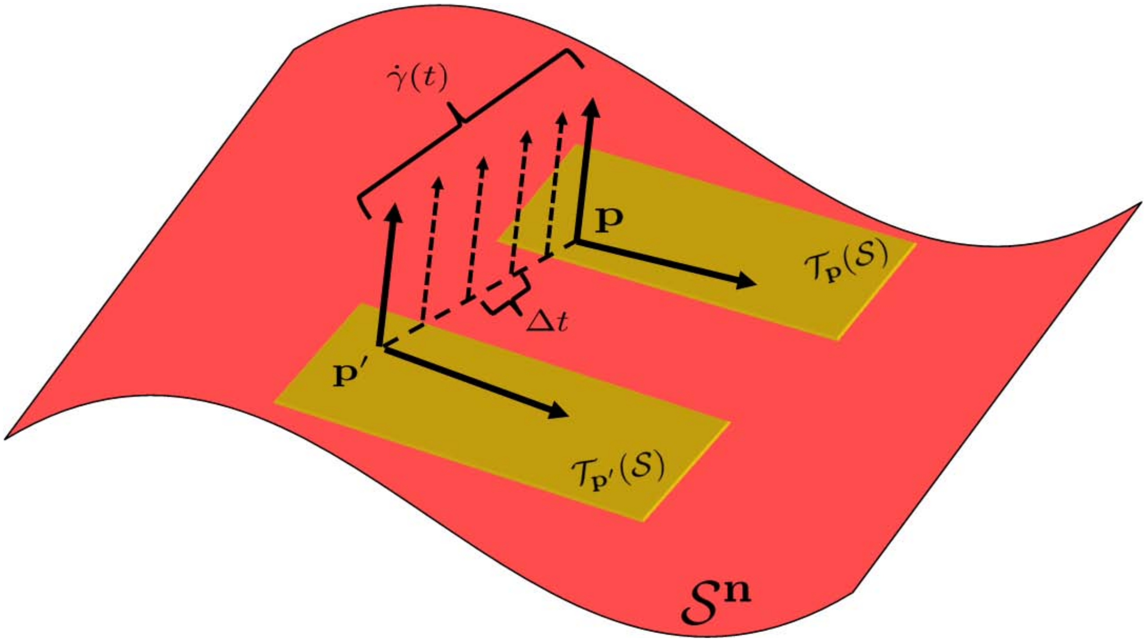

$\dot{\gamma}(t)$
$\mathbf{p}$
$\mathcal{T}_{\mathbf{p}}(\mathcal{S})$
Δt
$\mathbf{p}'$
$\mathcal{T}_{\mathbf{p}'}(\mathcal{S})$
$\mathcal{S}^{\mathbf{n}}$

Figure 2.3: The connection between tangent spaces $\mathcal{T}_{\mathbf{p}}(\mathcal{S})$ and $\mathcal{T}_{\mathbf{p}'}(\mathcal{S})$.

Having just presented the concept of motion or movement along the surface of a manifold $\mathcal{S}$, it is natural to now present the differential geometric tool known as geodesics. Geodesics are well suited for tasks involving motion along the surface of $\mathcal{S}$.

2.2.4 Geodesics. Considering the surface of a smooth manifold $\mathcal{S}$, if there exists a curve that passes through points $\mathbf{p}$ and $\mathbf{q}$ on $\mathcal{S}$ with the property that the tangent vectors along the curve are all parallel to one another, then curve γ is referred to as a geodesic. Generally speaking, the geodesic can be thought of as being analogous to straight lines in Euclidean space. Formally, geodesics on manifold $\mathcal{M}$ are defined as follows [42], [82].

Definition 2.2.10 (Geodesic)**.** *For any two points $\mathbf{p}$ and $\mathbf{p}'$ on the surface of the differentiable manifold $\mathcal{S}$ there will exist an infinite number of curves connecting the two points over a unit time interval defined as $t \in [0,1]$ and $\gamma(0) = \mathbf{p}$ and $\gamma(1) = \mathbf{p}'$. The curve that is characterized by the following equivalent properties:*

1. *$\dot{\gamma}(t)$ is constant for $t \in [0,1]$*
2. *$\ddot{\gamma}(t) = 0$ for all $t \in [0,1]$*
3. *All $\dot{\gamma}(t)$ with $t \in [0,1]$ are parallel.*

will be a length minimizing curve, or geodesic, between the two points $\mathbf{p}$ and $\mathbf{p}'$.

Figure 2.4 depicts the geodesic that passes through both points along the surface of $\mathcal{S}$ as a dashed line.

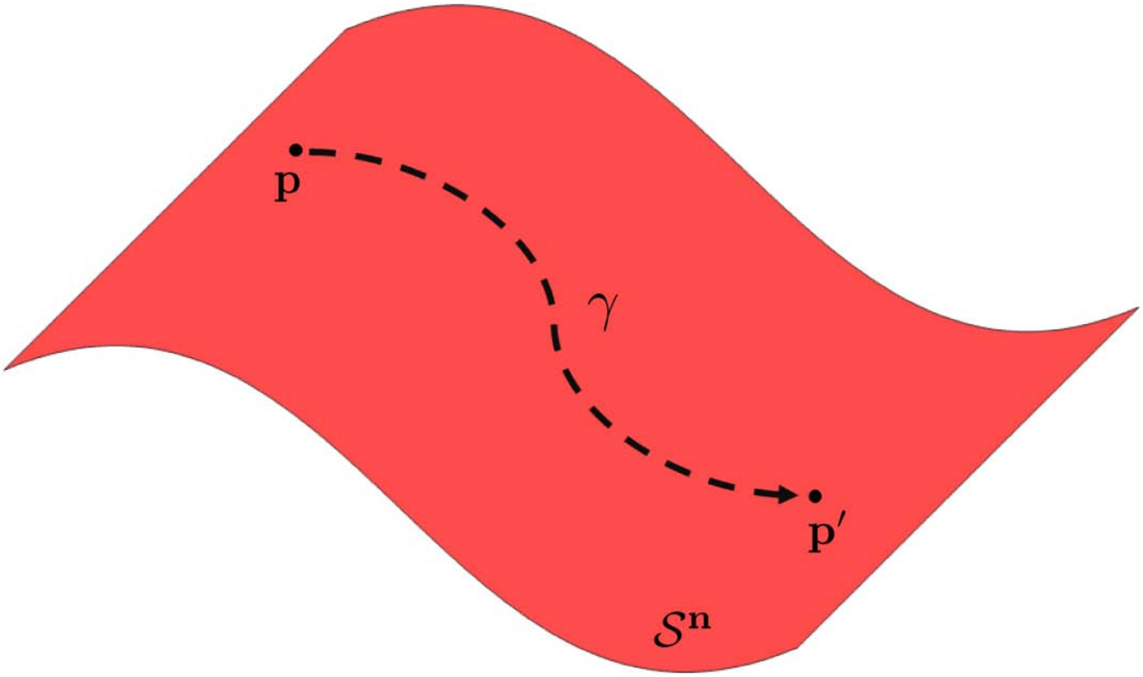


Figure 2.4: A geodesic curve γ along the surface of $\mathcal{S}$

2.2.5 Manifold Mapping Operators. A tangent vector can be interpreted as providing a sense of direction on the manifold $\mathcal{S}$. Now, it is not a big leap to suggest that since geodesics are a means of determining shortest length paths between two distinct points $\mathbf{p}$ and $\mathbf{q}$, and geodesics are defined uniquely by tangent vectors, that a natural means for examining propagation along the manifold $\mathcal{S}$ can be established. In fact, the above properties of geodesics lead directly to the definition of the manifold exponential operator or exponential mapping. The exponential mapping is defined as [42]:

Definition 2.2.11 (Exponential Map)**.** *Let* $\mathrm{ExpMap}_{\mathbf{p}}(\boldsymbol{\tau}) = \mathbf{p}(1)$*, that is, the image of* $\boldsymbol{\tau}$ *under the exponential mapping is defined to be the point on the unique geodesic defined by* $\boldsymbol{\tau}$ *such that the parameter takes on the value of* $(+1)$*, and stated more compactly by:*

$$\mathrm{ExpMap}_{\mathbf{p}} = \{\mathrm{ExpMap}_{\mathbf{p}}(\boldsymbol{\tau}) : \mathcal{T}_{\mathbf{p}}(\mathcal{S}) \to \mathcal{S} \mid \boldsymbol{\tau} \mapsto \gamma(1; \mathbf{p}, \boldsymbol{\tau})\}, \tag{2.3}$$

and is a one-to-one mapping between a neighborhood of point $\mathbf{p}$ *and the tangent space* $\mathcal{T}_{\mathbf{p}}(\mathcal{S})$*.*

The exponential mapping is a useful tool in establishing a relationship between Bayesian estimation and differential geometry given in Chapter III.

The inverse of the exponential map is the logarithmic map and is defined as

$$\mathrm{LogMap}_{\mathbf{p}}(\mathbf{q}) = \mathrm{ExpMap}_{\mathbf{p}}^{-1}(\boldsymbol{\tau}) = \boldsymbol{\tau}. \tag{2.4}$$

Note that the exponential and logarithmic mappings vary as the point $\mathbf{p}$ moves, and the specific forms of the exponential and logarithmic operators depends on the manifold that they are defined on. Figure 2.5 depicts the concepts of exponential and logarithmic mappings.

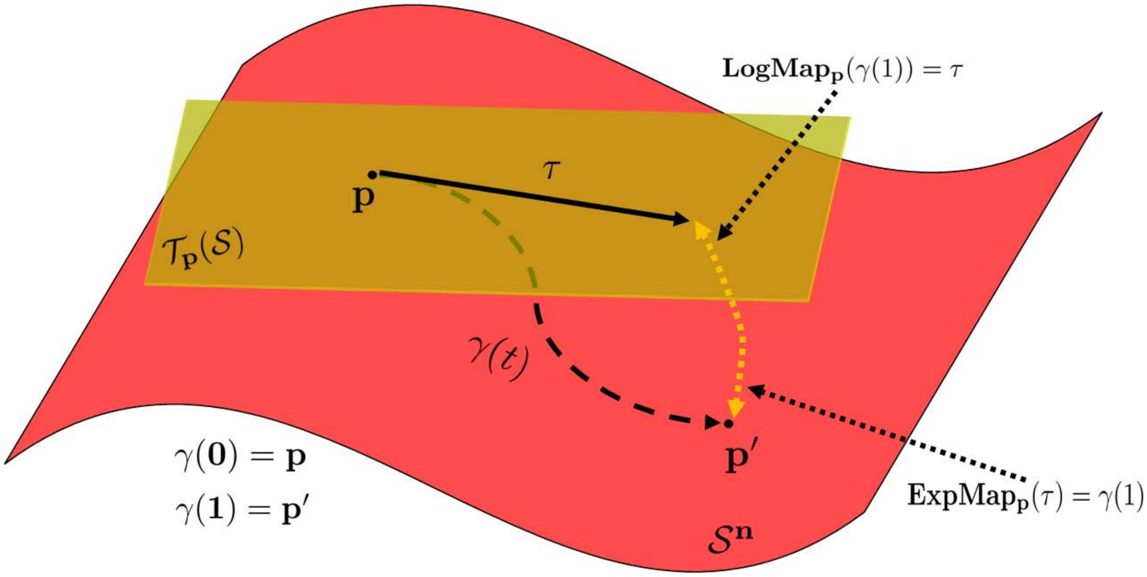

$\mathbf{LogMap_p}(\gamma(1)) = \tau$
τ
$\mathbf{p}$
$\mathcal{T}_{\mathbf{p}}(\mathcal{S})$
$\gamma(t)$
$\mathbf{p}'$
$\gamma(\mathbf{0}) = \mathbf{p}$
$\gamma(\mathbf{1}) = \mathbf{p}'$
$\mathbf{ExpMap_p}(\tau) = \gamma(1)$
$\mathcal{S}^{\mathbf{n}}$

Figure 2.5: Exponential and logarithmic mappings

2.2.6 Completeness Assumption. According to Pennec [226], the following is the definition of geodesic completeness:

Definition 2.2.12 (Geodesically Complete)**.** *A manifold is said to be geodesically complete if the domain, of definition $\mathcal{D}$ of all geodesics can be extended to the set of reals* $\mathbb{R}$.

From a practical perspective, the implication of Definition 2.2.12 is that there exists a geodesic that is length minimizing between any two points residing on the manifold, if a manifold is geodesically complete. Furthermore, the exponential map will be defined for all $\mathbf{p} \in \mathcal{S}$ and $\boldsymbol{\tau} \in \mathcal{T}_{\mathbf{p}}(\mathcal{S})$. Geodesic completeness is a natural segway to the Hopf-Rinow-De Rham Theorem [166], which is both practically useful and historically significant. The following theorem can be found in the text of Lee [166].

Theorem 2.2.1 (Hopf-Rinow-De Rham)**.** *A connected Riemannian manifold is geodesically complete if and only if it is complete as a metric space. Furthermore, on such a manifold there always exists at least one length minimizing geodesic that passes between any two points on the surface of the manifold.*

Throughout the remainder of this dissertation any discussion concerning manifolds will be under the assumption that the manifold is geodesically complete.

2.2.7 Comparison of Geometries. In general, standard mathematical operations that exist in Euclidean vector spaces like simple addition and subtraction do not exist on Riemannian manifolds. The role of such operations can be interpreted as being provided by operations such as exponential maps and logarithmic maps. To solidify this fact, Table 2.3.2, courtesy of [299], shows a comparison of the operations in a vector space and the corresponding operations on a general Riemannian manifold. The correspondences

Table 2.1: Relationship between arithmetic operations of addition and subtraction that are available in Euclidean vector spaces and the corresponding operations that are available on Riemannian manifolds. (After [299])

Operation	Euclidean Space	Riemannian Manifold
Subtraction	$\mathbf{x} = \mathbf{p} - \mathbf{q}$	$\mathbf{x} = \mathrm{LogMap}_{\mathbf{p}}(\mathbf{q})$
Addition	$\mathbf{p} = \mathbf{q} + \mathbf{x}$	$\mathbf{q} = \mathrm{ExpMap}_{\mathbf{p}}(\mathbf{x})$
Distance	$D(\mathbf{p}\Vert\mathbf{q}) = \Vert\mathbf{q} - \mathbf{p}\Vert$	$D(\mathbf{p}\Vert\mathbf{q}) = \Vert\mathbf{x}\Vert_{\mathbf{p}}$
Mean Value	$\sum_{i=1}^{n}(\mathbf{p}_i - \overline{\mathbf{p}}) = 0$	$\sum_{i=1}^{n}\mathbf{x}\mathbf{p}_i = 0$
Gradient Descent	$\mathbf{p}_{t+\varepsilon} = \mathbf{p}_t - \varepsilon\nabla\mathbf{Cut}\left(\mathbf{p}_t\right)$	$\mathbf{p}_{t+\varepsilon} = \mathrm{ExpMap}_{\mathbf{p}_t}(-\varepsilon\nabla\mathbf{Cut}\left(\mathbf{p}_t\right))$
Geodesic Interpolation	$\mathbf{p}(t) = \mathbf{p}_0 + t\mathbf{p}_0\mathbf{p}_1$	$\mathbf{p}(t) = \mathrm{ExpMap}_{\mathbf{p}_0}(t\mathbf{p}_0\mathbf{p}_1)$

shown in Table 2.3.2 are particularly useful in that they allow generalizing algorithms that are valid in vector spaces to Riemannian manifolds. A Riemannian manifold of particular interest in this dissertation is the unit sphere. The unit sphere has a well understood geometry, and this, along with a natural relationship with traditional Bayesian estimation theory, make it a valuable curved surface in the context of this dissertation. As a direct consequence, the unit sphere is given a thorough geometric description in the next Section.

2.3 The Unit Sphere: $\mathcal{S}^n$

The constructs that will be the most useful in this endeavor can be defined specifically for the manifold known as the unit hypersphere. A sphere is a well studied geometrical entity and serves as an illustrative manifold for several differential geometry presentations. It should be noted that throughout this section, the unit sphere associated with $\mathbb{R}^3$ will be used as a means of solidifying the concepts, but all topics mentioned are naturally extendable to higher dimensions.

2.3.1 Definitions & Relationships. An assumption that is made in this dissertation is that any two probability density functions that will be compared will reside on the same manifold and are *close enough* to make the analysis methods relevant and valid. The phrase *close enough* is somewhat vague and lacks any mathematical rigor. In an attempt to make more formal the definition we introduce the concept of injectivity radius. The injectivity radius denoted as $\mathbf{inj}\left(\mathbf{p}\right)$, is defined as [228]

Definition 2.3.1 (Injectivity Radius)**.** *The injectivity radius at a point $\mathbf{p} \in \mathcal{S}^n$ is the largest radius r such that the ball $\mathbf{B}\left(0, r\right) \subseteq \mathcal{T}_{\mathbf{p}}(\mathcal{S}^n)$ is an open ball, implying that the exponential mapping*

$$\mathrm{ExpMap}_{\mathbf{p}} : \mathbf{B}\left(0, r\right) \to \mathbf{B}\left(\mathbf{p}, r\right), \tag{2.5}$$

is a diffeomorphism.

Under the assumption that a manifold $\mathcal{S}^n$ is a geodesically complete Riemannian manifold, of interest here is the subset of tangent vectors such that the geodesic is defined as

$$\gamma(t) \mapsto \mathrm{ExpMap}_{\mathbf{p}}(t \cdot \boldsymbol{\tau}), \tag{2.6}$$

where t is used to denote the parameterization on the tangent vector. The geodesic defined in Equation (2.6) is the length minimizing geodesic up to $\gamma(t) = 1 + \varepsilon$ for $\varepsilon > 0$. If we denote the subset of tangent vectors to be $\mathbf{ST}$, an interesting result is the concept of cut locus. Formally, the cut locus is defined as [57]

Definition 2.3.2 (Cut Locus)**.** *If an exponential mapping is a diffeomorphism from* $\mathbf{ST}$ *onto its image defined as* $\mathrm{ExpMap}_{\mathbf{p}}(\mathbf{ST}) \in \mathcal{S}$*, then the portion of the geodesic that remains, (ie* $\mathcal{S} - \mathrm{ExpMap}_{\mathbf{p}}(\mathbf{ST})$*), is equal to* $\mathrm{ExpMap}_{\mathbf{p}}(\partial\mathbf{ST})$ *and is defined as the cut locus of* $\mathbf{p}$*, provided that the remaining set is not the empty set,* $\mathcal{S} - \mathrm{ExpMap}_{\mathbf{p}}(\mathbf{ST}) \neq \emptyset$*.*

Stated in a slightly different way, the maximum definition domain $\mathcal{D}$ where the exponential map is a diffeomorphism can be determined by setting $t \in [0, \infty)$ and calculating the geodesic such that it is a minimizing geodesic along it's entire path or up to some point $t_m < \infty$ and not for any point past t_m on the geodesic. The point t_m is known as a cut point. Given the Definition 2.3.2, a direct consequence is that the injectivity radius defined in 2.3.1 is now equal to the distance from a point $\mathbf{p}$ to the cut locus of point $\mathbf{p}$ and is defined to be

$$\begin{aligned} \mathbf{inj}\left(\mathbf{p}\right) &= D(\mathbf{p} \| \mathbf{Cut}\left(\mathbf{p}\right)) \\ &= \inf_{\mathbf{p'} \in \mathbf{Cut}(\mathbf{p})} D(\mathbf{p} \| \mathbf{p'}). \end{aligned} \tag{2.7}$$

Furthermore, the manifold $\mathcal{S}$ can be defined as

$$\mathcal{S} = \mathrm{ExpMap}_{\mathbf{p}}(\mathbf{ST}) \cup \mathbf{Cut}\left(\mathbf{p}\right). \tag{2.8}$$

Consider the following example using the unit Sphere $\mathcal{S}^n$. When regarding the unit sphere, the cut point is synonymous with the antipodal point and is defined on the unit hypersphere as

$$\mathbf{Cut}\left(\mathbf{p}\right) = \{-\mathbf{p}\} \text{ on } \mathcal{S}^n \tag{2.9}$$

The collection of all cut points of $\mathbf{p}$ along all geodesics is the cut locus $\mathbf{Cut}\left(\mathbf{p}\right)$. It should be noted that caution in representation is required. According to Pennec [226], the cut locus on the sphere can take on multiple representations due to the fact that there are an infinite number of geodesics that can originate at point $\mathbf{p}$ and end at $-\mathbf{p}$. This is due to the fact that each of the geodesics are determined through the exponential map of different tangent vectors.

2.3.2 Analytical Tools. Recall that geodesics were defined generically to be the path along the surface that connects two points and can be considered as a generalization of the concepts of lines and planes in $\mathbb{R}^2$ and $\mathbb{R}^3$ respectively. Consider the unit sphere in Figure 2.6 defined as

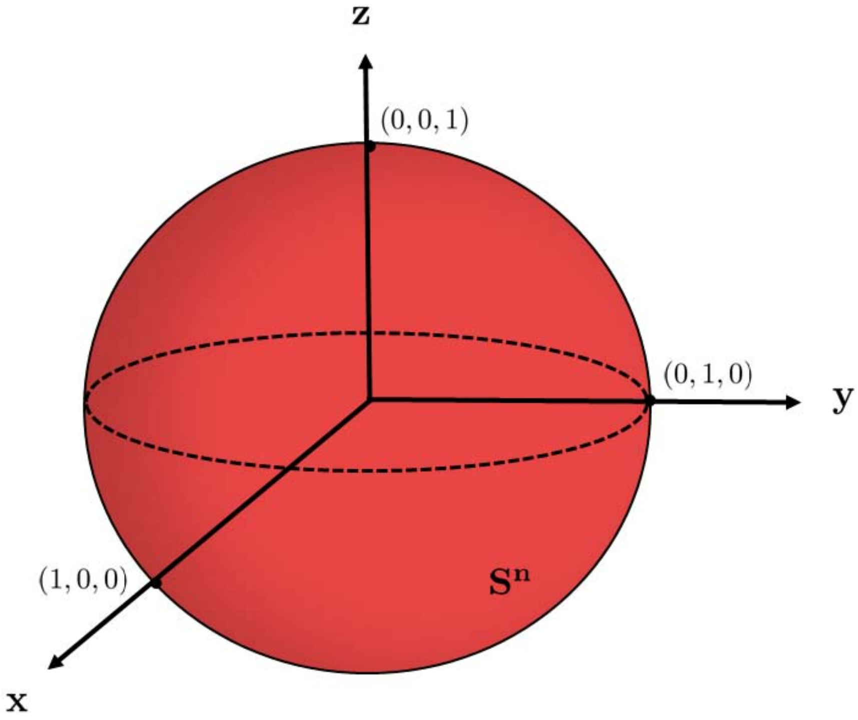


Figure 2.6: The unit sphere embedded in $\mathbb{R}^3$

$$\mathbb{S}^2 = \{(x, y, z) \in \mathbb{R}^3 \mid x^2 + y^2 + z^2 = 1\}. \tag{2.10}$$

The sphere defined in Equation (2.10) is actually a submanifold that is embedded in Euclidean $\mathbb{R}^3$ space. The following are the definitions for an immersion, an embedding, and a submanifold respectively, according to Do Carmo [82] and Warner [295]

Definition 2.3.3 (Immersion, Embedding, Submanifold)**.** *Let $\phi : \mathcal{S}^n \to \mathbb{R}^{n+1}$ be a C^∞ mapping. Then:*

1. *$\phi : \mathcal{S} \to \mathbb{R}$ is known as an immersion if $\partial\phi_{\mathbf{p}} : \mathcal{T}_{\mathbf{p}}(\mathcal{S}) \to \mathcal{T}_{\phi(p)}\mathbb{R}$ is injective for all elements $\mathbf{p} \in \mathcal{S}$.*
2. *If in addition ϕ is a homeomorphism onto $\phi(\mathcal{S}^n) \subset \mathbb{R}^{n+1}$, where $\phi(\mathcal{S}^n)$ has the subspace topology induced from $\mathbb{R}^{n+1}$, then ϕ is known as an embedding.*
3. *If $\mathcal{S}^n \subset \mathbb{R}^{n+1}$ and the inclusion $i : \mathcal{S} \subset \mathbb{R}$ is an embedding, then $\mathcal{S}^n$ is known as a submanifold of $\mathbb{R}^{n+1}$.*

The embedding of a unit hypersphere $\mathcal{S}^n$ into the larger $\mathbb{R}^{n+1}$ space leads to an intuitive interpretation of the tangent space $\mathcal{T}_{\mathbf{p}}(\mathcal{S}^n)$ located at any point $\mathbf{p}$ on $\mathcal{S}^n$, and is depicted in Figure 2.7. The tangent space is simply defined as

$$\mathcal{T}_{\mathbf{p}}(\mathcal{S}^n) = \{\boldsymbol{\tau} \in \mathbb{R}^{n+1} \mid \langle \boldsymbol{\tau} | \mathbf{p} \rangle = 0\} \tag{2.11}$$

where $\langle \boldsymbol{\tau} | \mathbf{p} \rangle$ is the usual inner product. The implication here is that the sphere possesses

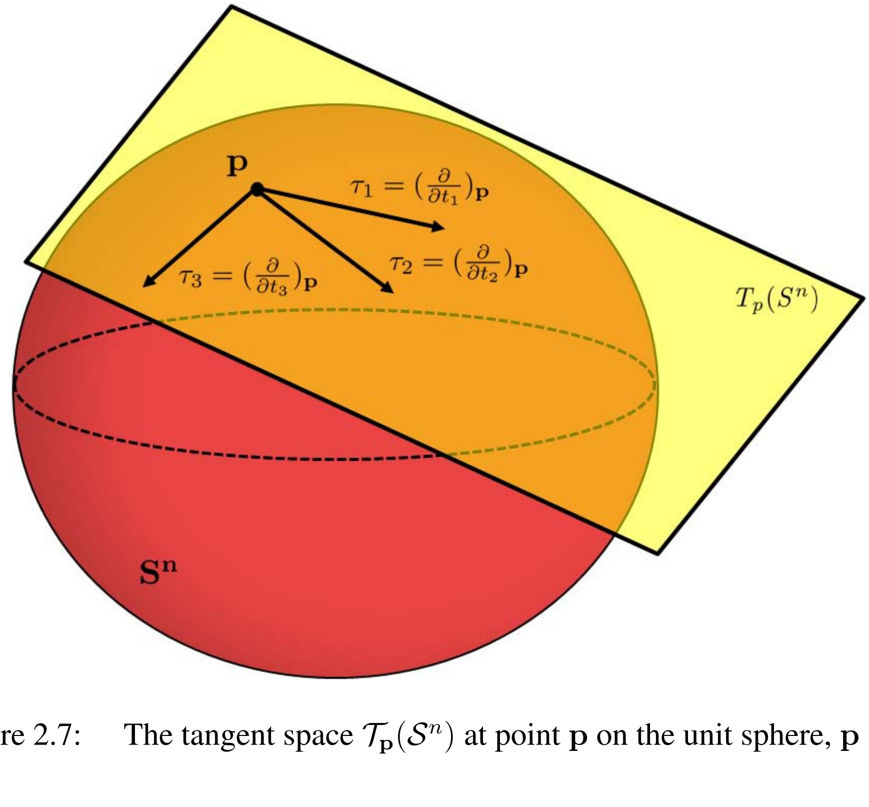

p
$\tau_1 = (\frac{\partial}{\partial t_1})_{\mathbf{p}}$
$\tau_2 = (\frac{\partial}{\partial t_2})_{\mathbf{p}}$
$\tau_3 = (\frac{\partial}{\partial t_3})_{\mathbf{p}}$
$T_p(S^n)$
$\mathbf{S^n}$

Figure 2.7: The tangent space $\mathcal{T}_{\mathbf{p}}(\mathcal{S}^n)$ at point $\mathbf{p}$ on the unit sphere, $\mathbf{p} \in \mathcal{S}^n$

a Riemannian metric by virtue of the embedding. The metric is a bilinear mapping

$$g_{\mathcal{S}^n} : \mathcal{T}_{\mathbf{p}}(\mathcal{S}^n) \times \mathcal{T}_{\mathbf{p}}(\mathcal{S}^n) \to \mathbb{R}^{n+1}, \tag{2.12}$$

and is defined for all points $\mathbf{p}$ residing on the sphere and is denoted by

$$g_{\mathcal{S}^n}(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2) = \langle \boldsymbol{\tau}_1 | \boldsymbol{\tau}_2 \rangle, \qquad \boldsymbol{\tau}_1, \boldsymbol{\tau}_2 \in \mathcal{S}^n, \tag{2.13}$$

with

$$\langle \boldsymbol{\tau}_1 | \boldsymbol{\tau}_2 \rangle = \boldsymbol{\tau}_1^T \boldsymbol{\tau}_2. \tag{2.14}$$

Additionally, given any two points $\mathbf{p}$ and $\mathbf{p'}$ on the surface of the sphere, the length of the geodesic connecting the two points can be determined according to

$$D(\mathbf{p} \| \mathbf{p'}) = \arccos(\langle \mathbf{p} | \mathbf{p'} \rangle). \tag{2.15}$$

Geodesics on the unit sphere $\mathcal{S}^n$ embedded in $\mathbb{R}^{n+1}$ are defined precisely by great circles [166], a segment of which is shown in Figure 2.8 connecting points $\mathbf{p}$ and $\mathbf{p'}$. Equation (2.15) can be used to describe elements that reside on the surface of the unit sphere in addition to the tangent vectors; a property that is not generally true if considering alternate parameterizations or surfaces. The reason is due to the fact that the unit sphere is actually embedded in $\mathbb{R}^n$, hence it inherits the typical notion of distance as defined in Euclidean space.

There are multiple expressions for the actual geodesic which depend on the choice of parameterization. A particularly useful parameterization is with respect to the direction of tangent vectors. Under this particular parameterization, the curve that connects two points on the surface of the sphere is the geodesic defined according to Equation (2.16) as

$$\gamma(t) = \cos(t)\mathbf{p} + \sin(t)\frac{\boldsymbol{\tau}}{\|\boldsymbol{\tau}\|}, \tag{2.16}$$

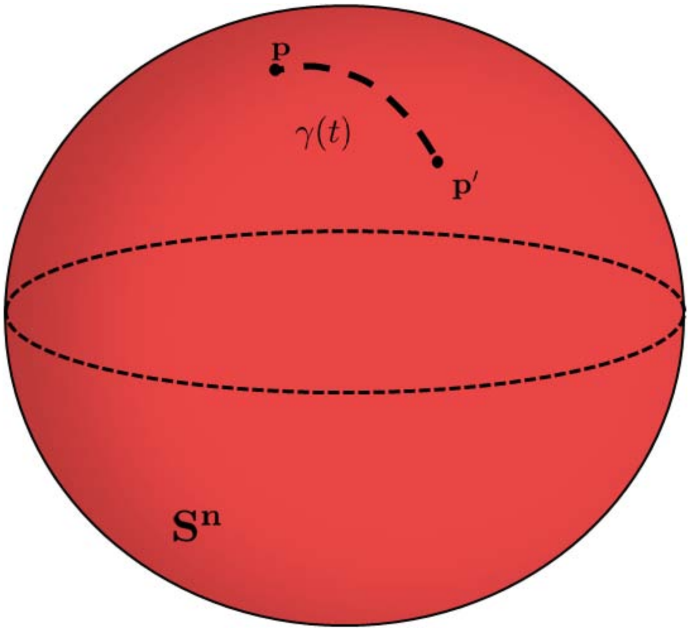


Figure 2.8: Geodesic $\gamma(t)$ connecting $\mathbf{p}$ and $\mathbf{p'}$ is a segment of a great circle on $\mathcal{S}^n$

where tangent vector $\boldsymbol{\tau} \in \mathcal{T}_{\mathbf{p}}(\mathcal{S}^n)$.

Recalling Equation (2.3), which defines the exponential map, and substituting Equation (2.16) yields a simple analytic expression for the exponential map, which is defined as

$$\mathrm{ExpMap}_{\mathbf{p}}(\boldsymbol{\tau}) = \cos(\|\boldsymbol{\tau}\|)\mathbf{p} + \sin(\|\boldsymbol{\tau}\|)\frac{\boldsymbol{\tau}}{\|\boldsymbol{\tau}\|}. \tag{2.17}$$

Finally, the logarithmic map defined in Equation (2.4) takes the geodesic with endpoint $\mathbf{p'}$ with respect to starting point $\mathbf{p}$ and maps to the unique tangent vector $\boldsymbol{\tau}$ that at $t = 0$ is tangent to point $\mathbf{p}$ in the direction of endpoint $\mathbf{p'}$ and has constant velocity over an unit interval. The logarithmic map is expressed as

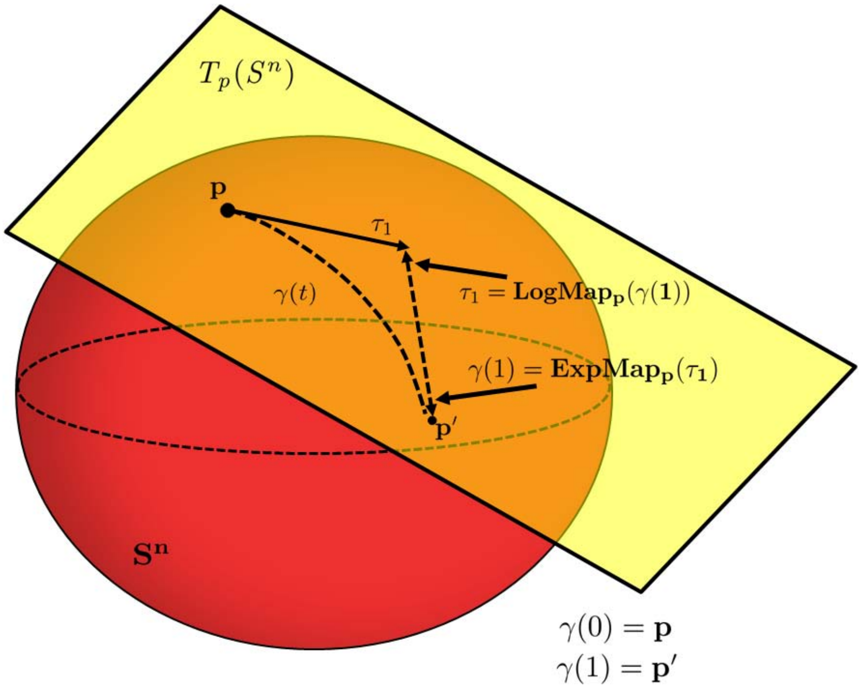
$$\mathrm{LogMap}_{\mathbf{p}}(\mathbf{p'}) = \boldsymbol{\tau}. \tag{2.18}$$

Equation (2.18) is calculated by the following two steps

$$\boldsymbol{\tau}_1 = \mathbf{p}' - \langle \mathbf{p}' | \mathbf{p} \rangle \mathbf{p} \tag{2.19}$$

$$\boldsymbol{\tau} = \frac{\boldsymbol{\tau}_1 \arccos(\langle \mathbf{p}' | \mathbf{p} \rangle)}{\sqrt{\langle \boldsymbol{\tau}_1 | \boldsymbol{\tau}_1 \rangle}} \tag{2.20}$$

Both the exponential map and the logarithmic map are shown in Figure 2.9. Equation (2.17) provides a means of calculating movement on the manifold $\mathcal{S}^n$ and Equations (2.19) and (2.20) provide a means of expressing the movement on the manifold in terms of tangent vectors in the tangent space. Table 2.3.2 summarizes the tools available for the unit n-Sphere [145].



$T_p(S^n)$
$\mathbf{p}$
τ_1
$\gamma(t)$
$\tau_1 = \mathbf{LogMap_p}(\gamma(\mathbf{1}))$
$\gamma(1) = \mathbf{ExpMap_p}(\tau_\mathbf{1})$
$\mathbf{p}'$
$\mathbf{S^n}$
$\gamma(0) = \mathbf{p}$
$\gamma(1) = \mathbf{p}'$

Figure 2.9: The exponential and logarithmic maps for the unit sphere $\mathcal{S}^n$

Table 2.2: Key tools that are available for working on the unit n-Sphere. This table is a partial replication of a table found in [145].

Name	Operations for Unit n-Sphere: $\mathcal{S}^n$
Elements	$\mathcal{S}^n = \{\boldsymbol{\tau} \in \mathbb{R}^{n+1} \mid \boldsymbol{\tau}_1^T \boldsymbol{\tau}_1 = 1\}$
Tangent Space	$\mathcal{T}_{\mathbf{p}}(\mathcal{M}) = \{\boldsymbol{\tau} \in \mathbb{R}^{n+1} \mid \boldsymbol{\tau}_1^T \boldsymbol{\tau}_2 \geqslant 0\}$
Projection Operator	$\mathbf{P}_{\mathbf{p}} : \mathbb{R}^{n+1} \rightarrow \mathcal{T}_{\mathbf{p}}(\mathcal{S}^n) : \boldsymbol{\tau} \mapsto \mathbf{P}_{\mathbf{p}}(\boldsymbol{\tau}) = \left(I - \mathbf{p}\mathbf{p}^T\right) \boldsymbol{\tau}$
Tangent Vector	$\boldsymbol{\tau} \ni \langle \boldsymbol{\tau} \vert \mathbf{p} \rangle = 0$
Inner Product	$\langle \mathbf{p} \vert \mathbf{p'} \rangle = \mathbf{p}^T \mathbf{p'} \in \mathbb{R}^{n+1}$
Vector Norm	$\Vert \boldsymbol{\tau} \Vert = \sqrt{\langle \boldsymbol{\tau} \vert \boldsymbol{\tau} \rangle}$
Distance	$D(\mathbf{p} \Vert \mathbf{p'}) = \arccos(\mathbf{p}^T \mathbf{p'})$
Exponential Map	$\mathrm{ExpMap}_{\mathbf{p}}(\boldsymbol{\tau}) = \cos(\Vert \boldsymbol{\tau} \Vert)\mathbf{p} + \sin(\Vert \boldsymbol{\tau} \Vert)\frac{\boldsymbol{\tau}}{\Vert \boldsymbol{\tau} \Vert}$
Logarithmic Map	$\mathrm{LogMap}_{\mathbf{p}}(\mathbf{p'}) = \frac{(\mathbf{p'} - \langle \mathbf{p'} \vert \mathbf{p} \rangle \mathbf{p}) \arccos(\langle \mathbf{p'} \vert \mathbf{p} \rangle)}{\sqrt{1 - \langle \mathbf{p'} \vert \mathbf{p} \rangle^2}}$
Curvature	$C = \frac{1}{r^2},$ (Unit Hypersphere: $\mathcal{S}^n$ C = 1)
Injectivity Radius	π
Convexity Radius	$\frac{\pi}{2}$
Cut Locus	$\mathbf{Cut}\,(\mathbf{p}) = \{-\mathbf{p}\}$ on $\mathcal{S}^n$

This concludes the presentation of differential geometry concepts. The next section is concerned with topics from data fusion models, architectures, and methods with an emphasis on particle filtering theory.

2.4 Data Fusion

Data fusion seems like a rather simple term to interpret and define. Data fusion methods play crucial roles within countless scientific disciplines. However, the wide spread use of data fusion methods has also led to more than a few challenges. For example, wide spread usage has been cited by various authors [172], [117] as the likely cause for the multiple definitions that can be found throughout the literature.

In general, as mentioned in Section 1.1, data fusion can be defined generically as the process of assimilating data from multiple sensors/sources in order to obtain a coherent and improved representation of what is being fused. Sources of data could originate from two or more sources collocated or spatially separated. The confusion resides in the vast range of environments that can be considered, and the equally as daunting volume of data sources available.

Measurement data obtained from real sensors will always be riddled with imperfections. Presumably data is collected for a purpose. Given a purpose and the presence of uncertainty in the data, a need often arises for delineating between the usable data and the corrupt data. The removal of uncertainty from data is often a task allocated to a data fusion process.

2.4.1 Common Models for Data Fusion . Models, in the context of Bayesian data fusion, are used to describe the physics that govern a particular process or measurement, and potentially their relationship. In a more general context, models are used as a means of defining, guiding, and potentially halting a particular procedure. The individual pieces of the overall data fusion model may consist of sensing tasks, processing tasks, decision making tasks, communication tasks, etc. Buried inside each one of these processes are numerous others that are required to be performed. Clearly, the level of complexity of a data fusion process can quickly become overwhelming if considered in its entirety. Nevertheless, there is need for such models and a few of the existing models are briefly described next.

In the mid 1980's, the U.S. Department of Defense (DoD) assembled a group of individuals and presented them with the task of developing a model for data fusion. The name of the group was the U.S. Joint Directors of Laboratories (JDL) data fusion group. In 1985 they published the original edition of the JDL data fusion model [294], and it is depicted in Figure 2.10. The primary need for the JDL model resided, at the time, within the DoD, due to a lack of standardized terminology and competing requirements that rendered agency to agency collaborations practically impossible [131].

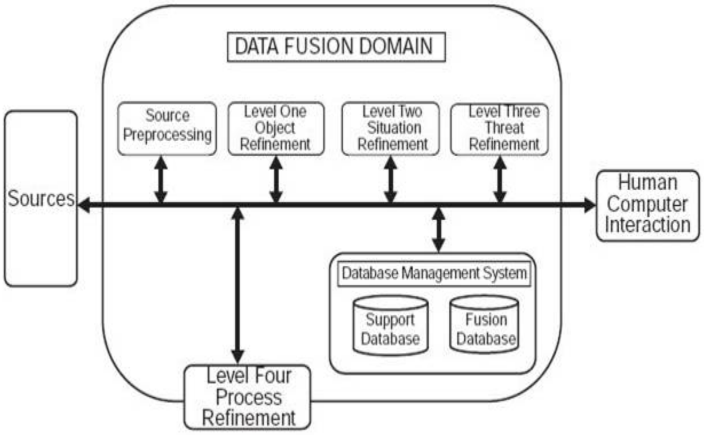

DATA FUSION DOMAIN
Source Preprocessing
Level One Object Refinement
Level Two Situation Refinement
Level Three Threat Refinement
Sources
Human Computer Interaction
Database Management System
Support Database
Fusion Database
Level Four Process Refinement

Figure 2.10: Data fusion model published by the Joint Directors of Laboratories (JDL) [294].

After the initial success of the first JDL model, several suggestions from multiple research communities were offered for improvement. In 1998 a 2nd edition was released, and by all accounts is still the most widely used data fusion model for functional descriptions and classification tasks [172], [267]. The revised model can be seen in Figure 2.11, [266].

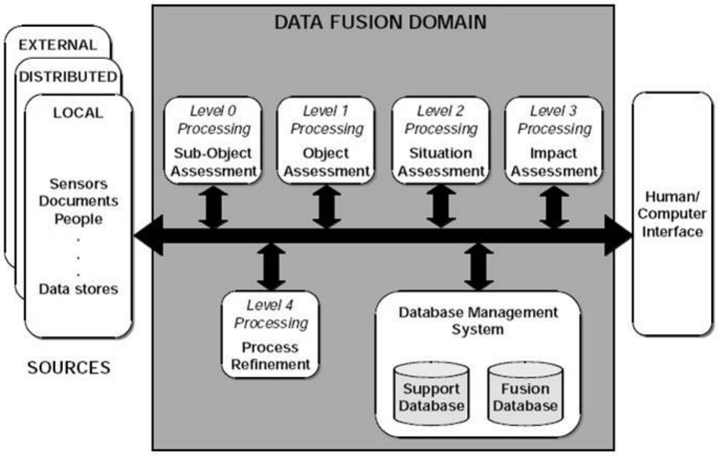


Figure 2.11: Revised data fusion model published by the JDL in 1998, and can be found in [266].

As can be seen in Figure 2.11 there exists a total of 5 levels at which fusion can take place within a process. These are: [116]

1. **Level 0: Sub-Object Assessment** - Fusion at the raw measurement level prior to any signal processing. Fusion occurs typically at the pixel or signal level. Level 0 offers an opportunity to separate and prioritize data from multiple sources.

2. **Level 1: Object Refinement** - Measurement-to-track association and state estimation. Level 1 is the level at which kinematic data gets fused and data association takes place.

3. **Level 2: Situation Refinement** - Object clustering or grouping in order to determine relationships. A higher level of data assessment than level 1 and usually involves heuristic analysis techniques.

4. **Level 3: Impact Assessment** - Threat assessment, estimation of intent, and prediction of consequences. The projection of the current assessment into the future to classify process options.

5. **Level 4: Process Refinement** - Resource management and adaptive processing. A decision maker resides at this level and monitors long term process health, identifies need information, and allocates resources based on information deficiencies.

Other models that can be found in the data fusion literature and are worthy of mention here include a model produced by Dasarathy [78], the waterfall model of Bedworth *et al.*, [190], Boyd's model [45], and the Omnibus model [30]. The interested reader is steered towards the associated references for further detail.

2.4.2 Non-Bayesian Data Fusion Methods. Traditionally, the tool that is most likely to be used for performing data fusion functions is Bayesian probability theory, as seen by the numerous Kalman filters, Kalman-like filter variants, and more recently particle filters found throughout the data fusion literature. However, additional tools have become available, including interval calculus [238], fuzzy logic [73], Dempster-Shafer (DST) or Evidence Theory [253], Neural Networks (NN) [290], fuzzy logic with the theory of possibility [303], Linear and Logarithmic Opinion Pools [2], category theory [270], and Dezert-Smarandache Theory (DSmT) [101]. There exist still more methods and techniques and new ones are regularly being published. Bayesian methods are chosen as a starting point in our endeavor and will be presented next.

2.4.3 Parametric Bayesian Data Fusion Methods. In the time since the original publication by Kalman [148] regarding his now famous filter, there have been numerous attempts to extend the original filter to address situations described by nonlinear models and/or non-Gaussian noises, to include [16, 27, 46, 58, 109, 128, 147, 192, 193, 271]. These references represent a small sampling of the more popular ones, and represent a good starting point for anyone interested in Bayesian estimation and nonlinear filtering techniques.

2.4.3.1 Extended Kalman Filter (EKF) Fusion. The extended Kalman filter, perhaps the most widely known technique, is based on approximations of the nonlinear functions used to describe the process model and/or measurement model. The extended Kalman filter has been implemented with varying degrees of success and is known to suffer from two significant drawbacks. First, the extended filter relies on a linear approximation by using a first order Taylor series expansion of the models. If the nonlinearities present in the system become severe enough and higher order disturbances begin impacting the filter's performance, filter divergence will be the likely result. The second issue is that there is a Gaussian assumption in the EKF framework that will likely become violated in some problems. The reason for the non-Gaussian disturbances is that when passed through a nonlinear model, there is no guarantee that a Gaussian noise disturbance will remain Gaussian.

Unlike the linear Kalman filter, there is no guarantee of bounded error or optimality accompanying the extended Kalman filter. The extended Kalman filter has the requirement that the Kalman gain and system covariance must be computed online. This is because they are functions that are dependent on the actual estimates and measurements. The extended Kalman filter may fail in situations where the system under consideration exhibits significant degrees of nonlinearity. Under low to moderate nonlinearities, the filter has been shown to yield reasonable results [237], [152], [208]. Finally, the extended Kalman filter requires that the nonlinearities under consideration be continuous. If they are not continuous, then this class of filter cannot be used [46]. In situations where this is the case the designer must consider alternatives. One such alternative gaining popularity is known as the unscented Kalman filter.

2.4.3.2 Unscented Kalman Filter (UKF) Fusion. The extended Kalman filter was premised on the fact that a suitable linear approximation for the system nonlinearities could be obtained. There is another possibility. Instead of trying to approximate the nonlinear functions through linearization techniques, what if the actual probability density function could be approximated [249]? This is the basis for the Unscented Kalman filter (UKF). Conceptually, the UKF approximates the probability density function with a set of deterministically chosen sample points which are transformed through the system nonlinearities [140]. The approximation technique is refereed to as the unscented transform (UT) [141].

The UT is a method of calculating the first two moments of a probability density function. The UT is based on the idea that “it is easier to approximate a probability density function than to approximate an arbitrary nonlinear function [137]." The suggestion here is that the state can be approximated by a set of deterministically chosen points in such a way that their sample mean and covariance faithfully represent the actual corresponding model state and covariance.

The following explanation is an intuitive description of the top level workings of the UKF. A system of nonlinear functions are used to propagate each sample point to yield a set of transformed points. Then the mean and covariance of the set of transformed samples are assumed to represent the mean and covariance of the filter states.

The basic UT may produce erroneous results in the situation that the number of state dimension exceeds 3. However, there are multiple examples within the data fusion literature of successful applications of the unscented Kalman filter with state dimensions that exceed 3. Some notable examples include [163], [165], and [194]. The most noticeable issue according to Šimandl [196], is due to the predicted measurement covariance and its ability to no longer be classified as positive definite. Many researchers have explored this problem and designed fixes including the scaled unscented transform [143], and the reduced UT [142]. Another alternative is based on Gauss-Hermite quadrature rule and was presented in [125].

However, the unscented Kalman filter is still tied to the Gaussian assumption. To fully relieve the constraints imposed by requiring Gaussian noise statistics and linearity requirements, yet still more alternatives need to be considered. In the context of decentralized data fusion, the following section highlights several popular methods currently in use, chief among them is a technique that has become known in the literature as particle filtering or sequential Monte Carlo (SMC) filtering.

2.5 Particle Filters

The Bayesian filtering framework is used to estimate some quantity of interest typically referred to as states. The Bayesian framework requires two probabilistic models. The first model represents how the state of a system will evolve over time and is typically called a process model [109] or propagation model [17]. The second model relates available noisy measurements to the states of interest and is often referred to as either a measurement model [192] or an observation model [52]. Additionally, the Bayesian approach imposes the added burden of requiring a prior probability be available upon initialization. The aforementioned burden rarely is a source of concern. Generally, the prior probability is formulated through either experience with the models being used, intuition regarding the scenario being considered, or some combination of these and/or additional factors. Once all of the required components are obtained, the goal of the Bayesian recursion is to calculate an estimate of the posterior density $\mathbf{p}(\mathbf{x}_k \mid \mathbf{Z}_k)$. The Bayesian filtering process is comprised of two steps, corresponding to the two required models mentioned previously. In a general manner of speaking, Equation (2.22) can be seen as an incorporation of the information about the prior state $\mathbf{x}_{k-1}$ available from the collection of measurements up to time $k-1$, denoted as $\mathbf{Z}_{k-1}$ in an attempt to predict the current state $\mathbf{x}_k$ prior to the incorporation of a measurement. Hence, a prediction or propagation step is performed using the process model and is realized with

$$\underbrace{\mathbf{p}(\mathbf{x}_k \mid \mathbf{Z}_{k-1})}_{\text{prior density}} = \int \mathbf{p}(\mathbf{x}_k \mid \mathbf{x}_{k-1}; \mathbf{Z}_{k-1})\mathbf{p}(\mathbf{x}_{k-1} \mid \mathbf{Z}_{k-1})\, d\mathbf{x}_{k-1} \qquad (2.21)$$

$$= \int \underbrace{\mathbf{p}(\mathbf{x}_k \mid \mathbf{x}_{k-1})}_{\text{transitional density}} \underbrace{\mathbf{p}(\mathbf{x}_{k-1} \mid \mathbf{Z}_{k-1})}_{\text{posterior density}}\, d\mathbf{x}_{k-1}. \qquad (2.22)$$

The second step corresponds to the measurement model that relates noisy measurements to states and is performed according to the following

$$\underbrace{\mathbf{p}(\mathbf{x}_k \mid \mathbf{Z}_k)}_{\text{posterior density}} = \frac{\mathbf{p}(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{Z}_{k-1})\mathbf{p}(\mathbf{x}_k \mid \mathbf{Z}_{k-1})}{\mathbf{p}(\mathbf{z}_k \mid \mathbf{Z}_{k-1})} \tag{2.23}$$

$$= \frac{\overbrace{\mathbf{p}(\mathbf{z}_k \mid \mathbf{x}_k)}^{likelihood}\mathbf{p}(\mathbf{x}_k \mid \mathbf{Z}_{k-1})}{\underbrace{\mathbf{p}(\mathbf{z}_k \mid \mathbf{Z}_{k-1})}_{evidence}}, \tag{2.24}$$

where

$$p(\mathbf{z}_k|\mathbf{Z}_{k-1}) = \int \mathbf{p}(\mathbf{z}_k|\mathbf{x}_k)\mathbf{p}(\mathbf{x}_k|\mathbf{Z}_{k-1})\, d\mathbf{x}_k. \tag{2.25}$$

Intuitively one can think of the update procedure as incorporating the evidence produced by the measurement. The evidence is used to *“adjust"* the posterior by the newly acquired data. Equations (2.22) and (2.24) form the basis for the Bayesian recursion.

Now, the ugly truth of the matter is that the recursive propagation of the posterior density, in most situations of practical interest, is simply not feasible. The primary reason is the need to solve multidimensional integrals that are usually only tractable in linear Gaussian systems [46], [27]. In fact, analytical solutions exist in only a handful of special cases, most notably when the models are linear and the noises are Gaussian, in which case the classical Kalman filter [148] provides the optimal solution. Hence, there arises a need to investigate filtering methods that can account for the nonlinearities and non-Gaussian noise disturbances, both of which often are needed to adequately describe realistic filtering scenarios. A viable method gaining more and more popularity in the data fusion literature is know as Sequential Monte Carlo (SMC) filtering or simply particle filtering.

Particle filtering techniques have seen a significant amount of research attention in the past decade. However, the beginnings of particle filtering can be traced back as far as the late 1940s. It was during this time that Nicolas Metropolis proposed studying dynamic systems by investigating the time evolution behaviors of a set of samples rather than focusing on individual samples [195, 257]. In the 1950s, sequential Monte Carlo techniques can again be found in the scientific literature [198], [133]. The 1970s saw the controls community make mention of Monte Carlo methods [118], [129], [7]. However, the use of monte carlo methods did not solicit much enthusiasm, mainly because of the lack of affordable computing power available at the time [79].

It wasn't until the seminal work presented in 1993 by Gordan *et al.,* [108] that particle filtering found mainstream use in engineering and science applications. The increase in particle filter related work over the past decade can be seen in the following survey articles [175], [20], [229], [55].

2.5.1 Monte Carlo Integration. A need often arises in nearly all fields of mathematics, engineering, and science to perform laborious and time consuming integrations. In fact, integration can be considered a foundational tool for any researcher in a technical field. Estimation is no exception when it comes to the need to be able to integrate. As is often the case, consider the task of evaluating a multidimensional integral, given generically in Equation (2.26)

$$I = \int g(\mathbf{x}) d\mathbf{x}, \qquad \mathbf{x} \in \mathbb{R}^n. \tag{2.26}$$

More often than not, integrals in the form given in Equation (2.26) will require a numerical method in order to evaluate the integral. In the context of estimation, a popular numerical method is the Monte Carlo approach to solving integrals. Essentially, the Monte Carlo approach to solving Equation (2.26) would be to factor the integrand such that

$$g(\mathbf{x}) = f(\mathbf{x})\mathbf{p}(\mathbf{x}), \tag{2.27}$$

with the conditions that

$$\mathbf{p}(\mathbf{x}) \geqslant 0 \quad \text{and} \qquad \int \mathbf{p}(\mathbf{x}) d\mathbf{x} = 1. \tag{2.28}$$

The new expression, $\mathbf{p}(\mathbf{x})$, in Equation (2.27) can be interpreted as a probability density function. Of value to the integration need is the fact that $\mathbf{p}(\mathbf{x})$ can be interpreted as a probability density function that can be sampled. The ability to sample from a probability density function is at the core of the Monte Carlo integration approach.

Monte Carlo integration capabilities provides distinct advantages over other numerical methods. First, to sample from a probability density function allows one to disregard multidimensional integration, and only consider their discrete counterparts in the form of algebraic sums [296]. Second, by virtue of incorporating a probability density function in the factorization of an integral, one can be assured that samples will be drawn from areas of high probability. Sampling from only high probability regions helps guard against wasted computation by ignoring regions of low probability, a luxury not afforded to continuous time integration. So, if a sufficiently large number of samples (*i.e.*, $N \gg 1$) can be drawn in accordance with the probability density $\mathbf{p}(\mathbf{x})$, then the Monte Carlo estimate of the integral takes the following form

$$I = \int f(\mathbf{x})\mathbf{p}(\mathbf{x})d\mathbf{x}, \tag{2.29}$$

which is merely the arithmetic mean of the samples. That is

$$I_N = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x})\delta(\mathbf{x} - \mathbf{x}^i), \tag{2.30}$$

where $\mathbf{x}^i$ is the i^{th} sample drawn in accordance with $\mathbf{p}(\mathbf{x})$. Figure 2.12 represents the Monte Carlo sample representation of a Gaussian probability density along with a continuous representation and the associated contours of the density for comparison.

An argument based on the strong law of large numbers can be made which, states that the average of the many *i.e.*, $N \to \infty$ independent random variables with a common

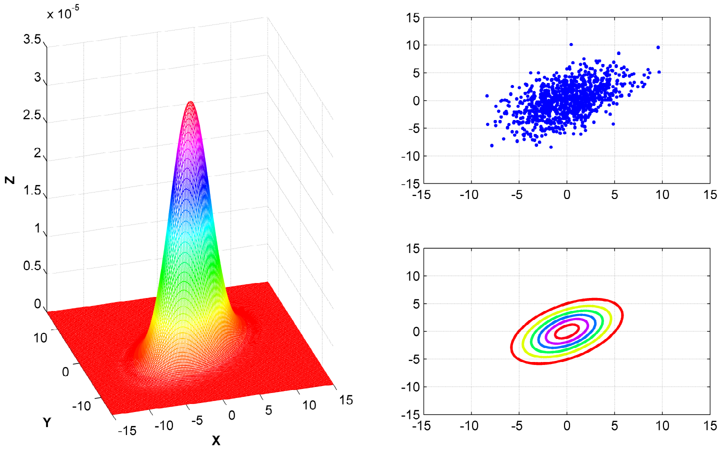


Figure 2.12: Gaussian probability density function, 1000 monte carlo samples, and associated contour plot.

mean and finite variance will converge to their common mean almost surely

$$
\begin{aligned}
I_N &= \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x})\delta(\mathbf{x}-\mathbf{x}^i) && (2.31)\\
&= \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^i) && (2.32)\\
&\xrightarrow[N\to\infty]{a.s.} \int f(\mathbf{x})\mathbf{p}(\mathbf{x})\,d\mathbf{x} && (2.33)
\end{aligned}
$$

where $a.s.$ means almost surely. Additionally, if the variance of $f(\mathbf{x})$ is finite

$$\sigma^2 = \int (f(\mathbf{x}) - I)^2 \mathbf{p}(\mathbf{x}) d\mathbf{x} < \infty \qquad (2.34)$$

then the central limit theorem holds, and the error in estimation converges in distribution to a zero-mean Gaussian as [16], [46]

$$\lim_{N\to\infty} \sqrt{N}(I_N - I) \to \mathcal{N}(0, \sigma^2) \qquad (2.35)$$

The error convergence can be attributed to the fact that the samples that are taken automatically come from regions in state space with high probability.

From the Bayesian estimation perspective the probability density $\mathbf{p}(\mathbf{x})$ can be interpreted as the posterior probability density [46]. The Bayesian approach to estimation often requires integration of high dimensional densities in order to render estimates. Unless the integral is tractable, which is rarely the case, numerical techniques are required for evaluating the integrands in the Bayesian recursion. Unfortunately, the number of samples to evaluate both $f(\mathbf{x})$ and $\mathbf{p}(\mathbf{x})$ increases dramatically with the dimensionality of the state space [46]. In response to the growth in required samples, it was stated previously that one of the benefits of choosing Monte Carlo integration techniques is that there is no longer a requirement to integrate over the entire space – just regions of high probability.

The above analysis is premised on the ability to sample from the density $\mathbf{p}(\mathbf{x})$ directly. In practice this is virtually impossible due to the nature of the density typically being multidimensional, non-Gaussian, and only known up to a constant of proportionality [46]. To overcome this fact, the technique of importance sampling can be employed.

2.5.2 Importance Sampling.

Importance sampling is a numerical technique that can be used to mitigate the impact of not being able to sample from a *true* underlying probability density. Generally speaking, importance sampling is a method of drawing samples from one density (often referred to as a proposal density or an importance density) in order to evaluate the expectation of another probability density by applying appropriate weighting. Importance sampling can be viewed as a generalization of the Monte Carlo integration previously presented [46], and is represented as a rewrite of Equation (2.26) such that

$$\begin{aligned} I &= \int g(\mathbf{x})d\mathbf{x} && (2.36)\\ &= \int f(\mathbf{x})\mathbf{p}(\mathbf{x})d\mathbf{x} && (2.37)\\ &= \int f(\mathbf{x})\frac{\mathbf{p}(\mathbf{x})}{\mathbf{q}(\mathbf{x})}\mathbf{q}(\mathbf{x})d\mathbf{x} && (2.38)\end{aligned}$$

such that

$$\int \mathbf{q}(\mathbf{x})d\mathbf{x} = 1, \qquad \text{and} \qquad \frac{\mathbf{p}(\mathbf{x})}{\mathbf{q}(\mathbf{x})} \qquad \text{is bounded above} \tag{2.39}$$

The probability density represented by $\mathbf{q}(\mathbf{x})$ is the so-called proposal density. Notice that $\mathbf{q}(\mathbf{x})$ is used to generate samples from $f(\mathbf{x})$ in a nonuniform manner. Sampling in this fashion facilitates the selection of samples from $f(\mathbf{x})$ with higher probabilistic implications. The authors in [46] say that the similarity between $f(\mathbf{x})$ and $\mathbf{q}(\mathbf{x})$ can be captured through the enforcement of the following constraint

$$f(\mathbf{x}) > 0 \Rightarrow q(\mathbf{x}) > 0, \qquad \forall \mathbf{x} \in \mathbb{R}^n. \tag{2.40}$$

Essentially, the condition in Equation (2.40) stipulates that samples taken from the proposal density $\mathbf{q}(\mathbf{x})$ will, at a minimum, be defined within the same portion of space, but can be defined over a larger region that encompasses the valid support domain of the function $f(\mathbf{x})$. This is also known as sharing a common sample support [304].

Within the Bayesian estimation framework, Monte Carlo samples generated from Equation (2.38), should consist of a large number of independent samples generated according to the proposal density. This will result in the weighted sum given by

$$I_N = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^{(i)})\tilde{\mathbf{w}}(\mathbf{x}^{(i)}), \tag{2.41}$$

where the weights $(\tilde{\mathbf{w}})$ are defined according to the quotient of density functions $\mathbf{p}(\mathbf{x}^{(i)})$ and $\mathbf{q}(\mathbf{x}^{(i)})$ *i.e.*,

$$\tilde{\mathbf{w}}(\mathbf{x}^{(i)}) = \frac{\mathbf{p}(\mathbf{x}^{(i)})}{\mathbf{q}(\mathbf{x}^{(i)})}, \tag{2.42}$$

and the tilde is used to denote the fact that the weights are not normalized. In order to actually calculate Equation (2.41), one will require access to the entire true or desired density $\mathbf{p}(\mathbf{x})$. Having access to $\mathbf{p}(\mathbf{x})$ in any practical situation of interest will likely not be the case. For this reason, the equality in Equation (2.42) should be replaced with proportionality, *i.e.*, [16]

$$\tilde{\mathbf{w}}(\mathbf{x}^{(i)}) \propto \frac{\mathbf{p}(\mathbf{x}^{(i)})}{\mathbf{q}(\mathbf{x}^{(i)})}. \tag{2.43}$$

Although subtle, this fact needs to be addressed, otherwise the whole concept of importance sampling becomes invalid. After all, sampling from a probability density that isn't the true probability density will produce invalid importance weights [46]. One must now consider if there exists a meaningful way, in which the importance weights should be normalized? The short answer is yes.

First, consider the fact that Equation (2.36) can be rewritten into the following form

$$I = \frac{\int g(\mathbf{x})\tilde{\mathbf{w}}(\mathbf{x})\mathbf{q}(\mathbf{x})d\mathbf{x}}{\int \tilde{\mathbf{w}}(\mathbf{x})\mathbf{q}(\mathbf{x})d\mathbf{x}}, \tag{2.44}$$

since

$$\int \tilde{\mathbf{w}}(\mathbf{x})\mathbf{q}(\mathbf{x})d\mathbf{x} \;=\; \int \mathbf{p}(\mathbf{x})d\mathbf{x} \;=\; 1. \tag{2.45}$$

Now, take the samples generated from using the proposal density, the corresponding importance weights, and substitute them into Equation (2.44) to obtain

$$\begin{aligned} I_N &= \frac{\displaystyle\sum_{i=1}^{N}\tilde{\mathbf{w}}(\mathbf{x}^{(i)})g(\mathbf{x}^{(i)})}{\displaystyle\sum_{j=1}^{N}\tilde{\mathbf{w}}(\mathbf{x}^{(j)})} & (2.46)\\ &= \sum_{i=1}^{N}\mathbf{w}(\mathbf{x}^{(i)})g(\mathbf{x}^{(i)}). & (2.47)\end{aligned}$$

The weights represented in Equation (2.47) can now be normalized. The normalization procedure is accomplished according to

$$\mathbf{w}(\mathbf{x}^{(i)}) = \frac{\tilde{\mathbf{w}}(\mathbf{x}^{(i)})}{\displaystyle\sum_{i=1}^{n}\tilde{\mathbf{w}}(\mathbf{x}^{(i)})} \tag{2.48}$$

Figures 2.13, 2.14, and 2.15 provide an illustration of the importance sampling process and the increase in approximation accuracy associated with the increase in the number of samples taken. With regards to Figures 2.13, 2.14, and 2.15, the goal is to try and approximate a Gaussian probability density function through the use of importance sampling. The green probability density is a uniform density defined on the interval ± 4, and is used as the proposal or importance density.

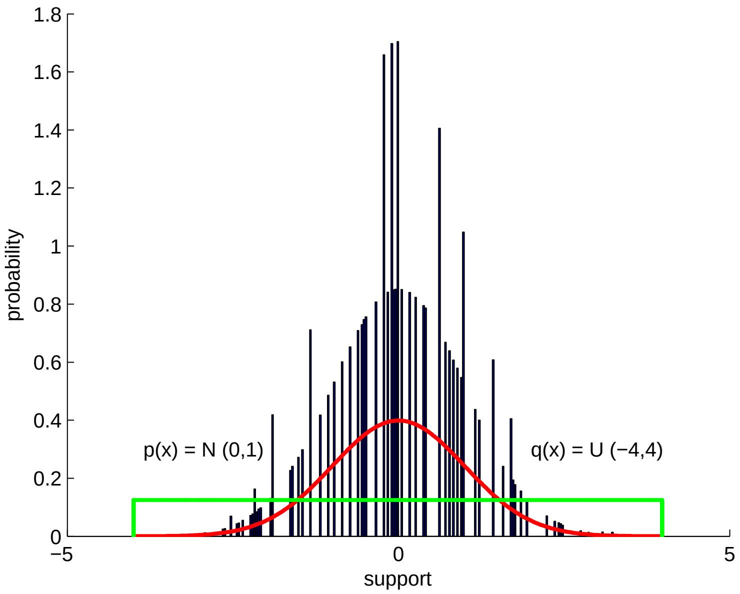


Figure 2.13: Importance sampling with 100 samples.

2.5.3 Sequential Importance Sampling (SIS). Importance sampling was just shown to address the issue of not being able to sample directly from a so-called true density. The practical issue with the presentation to this point is that importance sampling has been cast as a sort of batch estimation method, in the Bayesian sense. The batch interpretation stems from the fact the collection importance weights must be recalculated whenever a new measurement is to be used. In an attempt to extend the method of importance sampling to cases where it is desired to be able to formulate a recursive estimation scheme, Sequential Importance Sampling (SIS) has been developed to this end.

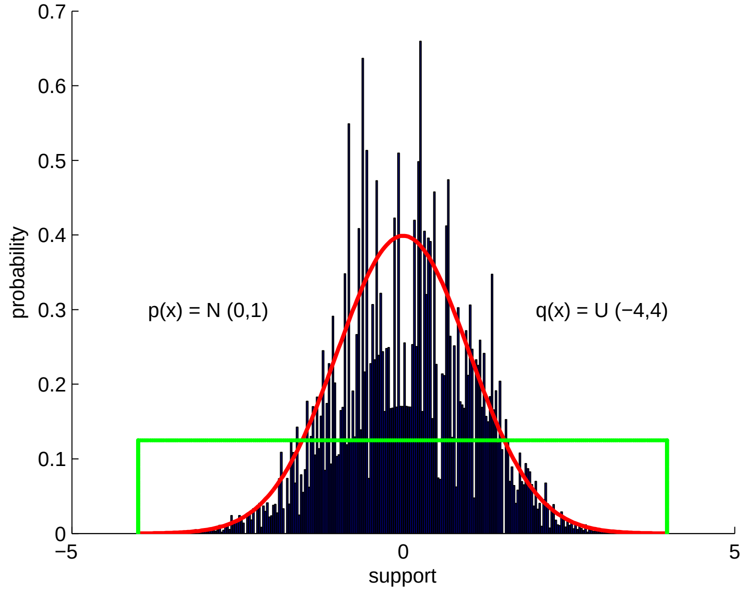


Figure 2.14: Importance sampling with 1000 samples.

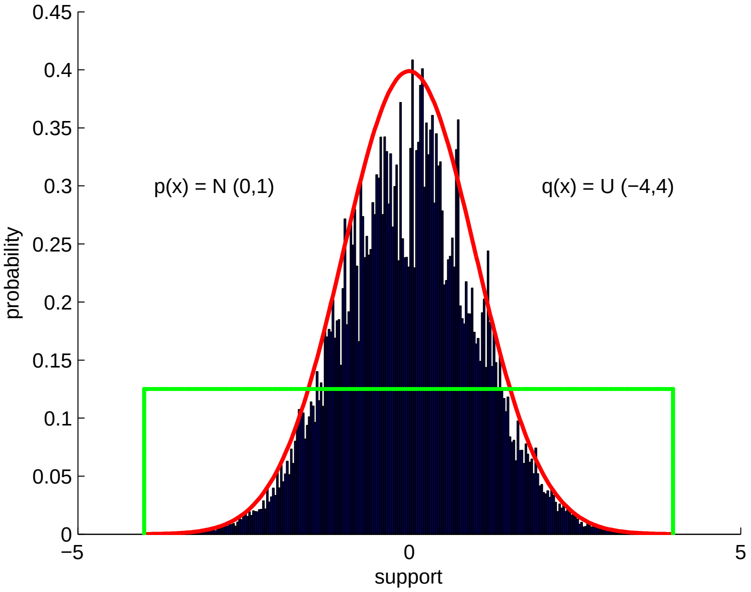


Figure 2.15: Importance sampling with 10000 samples.

In SIS, the purpose is to derive an estimate of the posterior density $\mathbf{p}(\mathbf{x}_k \mid \mathbf{z}_k)$ using the prior density $\mathbf{p}(\mathbf{x}_{k-1} \mid \mathbf{z}_{k-1})$ and the new measurement $\mathbf{z}_k$. In particle filtering terms, the goal is to produce new samples and their associated weights using the old samples and old weights [46].

Recall that one of the benefits to particle filtering approaches is the lack of restrictions placed upon the process and measurement models. Restrictions in the form of the necessity for linear filtering models and/or stochastic disturbances having to be described solely by Gaussian statistics. So the only way to represent the posterior density in a Bayesian framework is through the following iteration

$$\begin{aligned}
\mathbf{p}(\mathbf{X}_k \mid \mathbf{Z}_k) &= \frac{\mathbf{p}(\mathbf{z}_k \mid \mathbf{X}_k, \mathbf{Z}_{k-1})\mathbf{p}(\mathbf{X}_k \mid \mathbf{X}_{k-1})}{\mathbf{p}(\mathbf{z}_k \mid \mathbf{Z}_{k-1})} && (2.49)\\
&= \frac{\mathbf{p}(\mathbf{z}_k \mid \mathbf{X}_k, \mathbf{Z}_{k-1})\mathbf{p}(\mathbf{x}_k \mid \mathbf{X}_{k-1}, \mathbf{Z}_{k-1})\mathbf{p}(\mathbf{X}_{k-1}|\mathbf{Z}_{k-1})}{\mathbf{p}(\mathbf{z}_k \mid \mathbf{Z}_{k-1})} && (2.50)\\
&= \frac{\mathbf{p}(\mathbf{z}_k \mid \mathbf{x}_k)\mathbf{p}(\mathbf{x}_k \mid \mathbf{x}_{k-1})}{\mathbf{p}(\mathbf{z}_k \mid \mathbf{Z}_{k-1})}\mathbf{p}(\mathbf{X}_{k-1} \mid \mathbf{Z}_{k-1}), && (2.51)
\end{aligned}$$

where the use of the capital letters $\mathbf{X}$ and $\mathbf{Z}$ denotes the entire time history up to and including the indicated time iteration of states or measurements respectively. The denominator $\mathbf{p}(\mathbf{z}_k \mid \mathbf{Z}_{k-1})$ can be viewed as just a normalizing constant such that the expression in Equation (2.51) can be written according to

$$\mathbf{p}(\mathbf{X}_k \mid \mathbf{Z}_k) \propto \mathbf{p}(\mathbf{z}_k \mid \mathbf{x}_k)\mathbf{p}(\mathbf{x}_k \mid \mathbf{x}_{k-1})\mathbf{p}(\mathbf{X}_{k-1} \mid \mathbf{Z}_{k-1}). \tag{2.52}$$

The key assumption to make this algorithm legitimate is that the proposal density adheres to the following form [16], [46]

$$\mathbf{q}(\mathbf{X}_k \mid \mathbf{Z}_k) = \mathbf{q}(\mathbf{x}_k \mid \mathbf{X}_{k-1}, \mathbf{Z}_k)\mathbf{q}(\mathbf{X}_{k-1} \mid \mathbf{Z}_{k-1}). \tag{2.53}$$

This assumption is not particularly unrealistic, nor is it unreasonably restrictive. The following interpretation of the assumption expressed in Equation (2.53) is provided. Equation

(2.53) simply suggests that the state at time k and older are independent of the measurement at time $(k-1)$.

The SIS version of a particle filter attempts to provide estimates of the posterior density through a large number of samples $N \gg 1$. If the samples can be taken from the posterior, then an expression for the posterior estimate can be given as

$$\mathbf{p}(\mathbf{x}_k \mid \mathbf{Z}_k) \approx \sum_{i=1}^{N} \mathbf{w}(\mathbf{x}_k^{(i)})\delta(\mathbf{x}_k - \mathbf{x}_k^{(i)}). \tag{2.54}$$

In the unlikely situation that all of the samples are generated from the true probability density, each of the weights given in Equation (2.54) should all be set to one, which implies that all of the samples are equally likely. Finally, according to the law of total probability, which states that the sum of the weights must equal 1, the estimate in Equation (2.54) should be multiplied by $\frac{1}{N}$, where N is the total number of samples.

If the samples $\mathbf{x}_k^{(i)}$ in Equation (2.54) were drawn from the proposal density instead of the posterior density, which is likely the case, then according to Equation (2.42) the weights can be expressed in the form

$$\mathbf{w}_k^{(i)} \propto \frac{\mathbf{p}(\mathbf{x}_k^{(i)} \mid \mathbf{z}_k)}{\mathbf{q}(\mathbf{x}_k^{(i)} \mid \mathbf{z}_k)}. \tag{2.55}$$

With the availability of Equations (2.52) and (2.53), the development of the weight update given in Equation (2.55) can be derived according to the following method

$$\begin{aligned}
\mathbf{w}_k^{(i)} &\propto \frac{\mathbf{p}(\mathbf{x}_k^{(i)} \mid \mathbf{z}_k)}{\mathbf{q}(\mathbf{x}_k^{(i)} \mid \mathbf{z}_k)} && (2.56)\\
&\propto \frac{\mathbf{p}(\mathbf{z}_k \mid \mathbf{x}_k^{(i)})\mathbf{p}(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{k-1}^{(i)})\mathbf{p}(\mathbf{X}_{k-1}^{(i)} \mid \mathbf{Z}_{k-1})}{\mathbf{q}(\mathbf{x}_k^{(i)} \mid \mathbf{X}_{k-1}^{(i)}, \mathbf{Z}_k)\mathbf{q}(\mathbf{X}_{k-1}^{(i)} \mid \mathbf{Z}_{k-1})} && (2.57)\\
&= \mathbf{w}_{k-1}^{(i)} \frac{\mathbf{p}(\mathbf{z}_k \mid \mathbf{x}_k^{(i)})\mathbf{p}(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{k-1}^{(i)})}{\mathbf{q}(\mathbf{x}_k^{(i)} \mid \mathbf{X}_{k-1}^{(i)}, \mathbf{Z}_k)}. && (2.58)
\end{aligned}$$

The ability to recursively update the importance weights can be can now be realized, Furthermore, the ability to recursively update the importance weights permits the rewriting of Equation (2.54) in the following manner

$$\mathbf{p}(\mathbf{x}_k \mid \mathbf{z}_k) \approx \sum_{i=1}^{N} \mathbf{w}_k^{(i)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)}), \tag{2.59}$$

where the weights are normalized according to Equation (2.47).

2.5.4 Resampling. The sequential importance sampling algorithm just presented forms the basis for a generic particle filter. However, the algorithm suffers from some real drawbacks that make it unpractical in its present form. For example, since the particles are allowed to evolve over a time horizon, they will tend to spread out. The result of the temporal propagation of particles is an ever increasing particle variance. This means that all but a few samples will have a weight of zero, and will not contribute to the estimation of the desired density. The ramifications of an ever growing particle variance can be seen in both the computational burden and estimation accuracy of the particle filter. First, the basic algorithm is forced to propagate zero-weighted samples and the few significant samples can, at best, provide a very crude estimate of the posterior density. This phenomenon is known as *particle degeneracy* and is illustrated in Figure 2.16. Figure 2.16 is used to demonstrate just how particle degeneracy occurs. Figure 2.16 was generated as the result of implementing a sequential importance sampling algorithm, with an initial collection of particle numbering 1000. The original particle collection was uniformly distributed throughout a 2D space defined with domain $[0, 1]$ and range $[0, 1]$. The algorithm, as shown, underwent 400 particle propagations. At every iteration, the current collection of particles was randomly sampled according to a uniform distribution. The phenomenon of particle degeneracy is clearly evident in this figure. Notice, that the total number of particles went from an original collection size of 1000 to a final collection size of 5. To counter this problem a resampling step was proposed by [1].

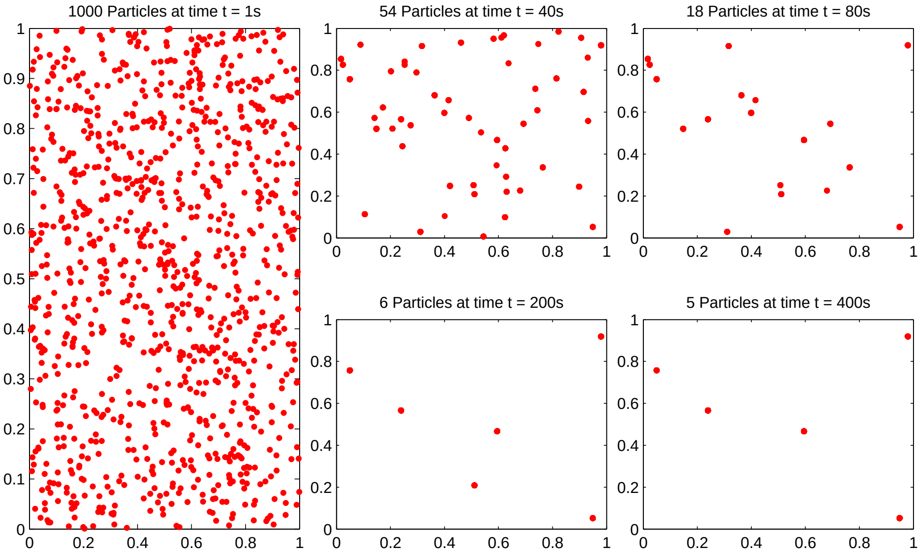


Figure 2.16: Illustration of the phenomenon known as particle degeneracy.

Resampling is an attempt to counter the degeneracy issue that plagues the SIS algorithm. One method to resampling was proposed by [7] and later in [1] involves monitoring the effective sample size. The effective sample size involves comparing covariances between samples from importance sampling and samples drawn from the posterior. This will provide a measure of how efficient the sampling is. The covariance comparison leads to an approximation of the effective sample size and an expression can be found in [175] and [33] and is given as

$$\hat{N}_{eff} \approx \frac{1}{\displaystyle\sum_{i=1}^{N} \left(\mathbf{w}_k^{(i)}\right)^2} \tag{2.60}$$

where $\mathbf{w}_k^{(i)}$ are the normalized weights that were calculated in Equation (2.58). The effective sample size will be bounded from below and above such that if all of the weights are equal, then the effective number of samples will be

$$N_{eff} = N \tag{2.61}$$

and if there is some natural number j such that $\mathbf{w}_k^{(j)} = 1$ and $\mathbf{w}_k^{(i)} = 0$ for all $i \neq j$ then [46]

$$N_{eff} = 1 \tag{2.62}$$

leading to

$$1 \leqslant N_{eff} \leqslant N \tag{2.63}$$

Now a decision rule needs to be implemented to determine when resampling is necessary based on Equation (2.60). It is unclear where the following resampling rule was first presented, but it has been suggested that resampling should be conducted if a threshold of

$$N_{th} = \frac{2N}{3} \tag{2.64}$$

is exceeded [19], [33]. When the threshold in Equation (2.64) is reached, the authors in [19] and [46] suggest drawing N new samples $\mathbf{x}_k^{(i)}$ from the approximated posterior given in Equation (2.59) such that the probability of choosing $\mathbf{x}_k^{(i)}$ is $\mathbf{w}_k^{(i)}$. Since the weights came from the estimated posterior, all of the new weights associated with the new samples should be set to $\frac{1}{N}$. This will effectively eliminate samples with low importance and multiply samples that have a greater contribution to the estimate so that the new “cloud" of particles are concentrated in the regions of state space with the most interest. Notionally, the selection of new particles is illustrated in Figure 2.17. In Figure 2.17, there were originally 15 samples with initial weights. The initial weights of the 15 samples, along with the resulting samples as a result of resampling are given explicitly in Table 2.5.4 and shown in Figure 2.17.

Table 2.3: Initial weights for a collection of particles and the particles that are generated as a result of resampling

Initial Weight	2	1	1	6	3	9	3	1	2	1	2	3	7	2	1
Resampled Particles	0	0	0	2	0	5	1	0	2	1	0	0	1	2	1

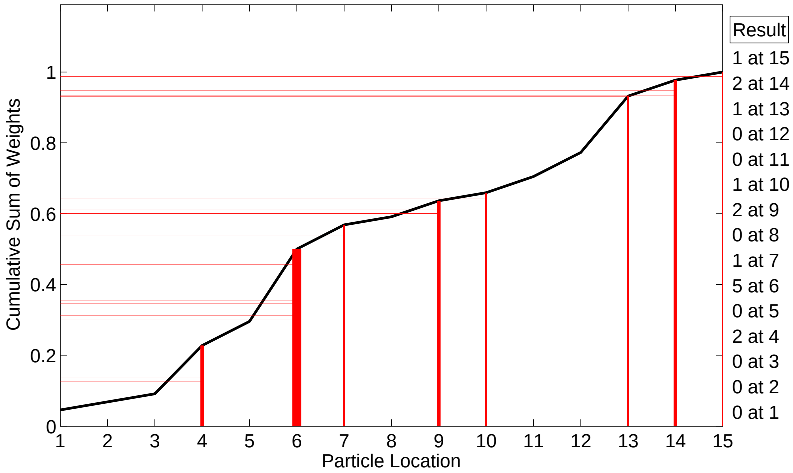


Figure 2.17: Demonstration of the resampling process. (figure adapted from [278])

Resampling, however, is not a panacea – there is a price to be paid. Statistically, the samples that are multiplied are no longer independent, since some of the samples are mere images of the originals, and resampling will always increase the variance on any estimate of the posterior density. Resampling, although necessary, needs to be implemented with some thought. The first attempt to resampling involved resampling at a fixed interval. If resampling was conducted at every iteration the filter formulation was coined the bootstrap method [108]. The method of fixed interval resampling has two distinct shortcomings. First, the covariance of the estimate is sometimes unnecessarily inflated due to resampling when it may not be required. Second, the determination of the actual resampling interval can only be determined by tedious trial and error.

There have been several suggested alternative resampling algorithms in the particle filtering literature. In fact the authors in [235] present a comparison of a few selected resampling schemes. Additionally, references [175], [16], and [153] also provide presentations on resampling techniques.

2.5.5 Sample Importance Resampling (SIR). The sample importance resampling algorithm for particle filtering is actually a special case of the sequential importance sampling algorithm. The sample importance resampling algorithm was first proposed in [108] as an attempt to address the known problem of particle degeneracy. The algorithm is easily implemented, as is the SIS algorithm. The crucial step in the design of the SIR algorithm is the choice of proposal density. It has been suggested by the authors of [175] and [16] that the optimal choice for a proposal density is one that minimizes the variance of the importance weights and was shown in [46], [175], and [84] to be

$$\mathbf{q}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)_{opt} = \mathbf{p}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k) = \frac{\mathbf{p}(\mathbf{z}_k|\mathbf{x}_k, \mathbf{x}_{k-1}^{(i)})\mathbf{p}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)})}{\mathbf{p}(\mathbf{z}_k|\mathbf{x}_{k-1}^{(i)})}. \tag{2.65}$$

However, sampling from this proposal density is impractical for any arbitrary density. Instead, the authors of [16] suggest sampling from the transitional prior density, *i.e.,*

$$\mathbf{q}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k) = \mathbf{p}(\mathbf{x}_k|\mathbf{x}_{k-1}). \tag{2.66}$$

This proposal density actually has samples drawn in the form of

$$\mathbf{x}_k^{(i)} \approx \mathbf{p}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)}). \tag{2.67}$$

In order to generate a sample $\mathbf{x}_k^{(i)}$ one must first generate a sample from the process noise $\omega_{k-1}^{(i)} \approx \mathbf{p}_\omega(\omega_{k-1})$ where $\mathbf{p}_\omega$ is the probability density function of the process noise. Then, the samples and process noise are propagated through the system nonlinear dynamics to update the samples, *i.e.,*

$$\mathbf{x}_k^{(i)} = f\left(\mathbf{x}_{k-1}^{(i)}, \mathbf{w}_{k-1}^{(i)}\right). \tag{2.68}$$

Given the choice of proposal density, up to a constant of proportionality, in Equation (2.66), the update equation for the importance weights can be expressed according to the following equation

$$\mathbf{w}_k^{(i)} \propto \mathbf{w}_{k-1}^{(i)}\mathbf{p}(\mathbf{z}_k|\mathbf{x}_k^{(i)}). \tag{2.69}$$

It should be noted that in the original sample importance resample algorithm the resampling step was set to occur with every iteration of the algorithm. This means that all of the prior weights will be forced to be

$$\mathbf{w}_{k-1}^{(i)} = \frac{1}{N}. \tag{2.70}$$

Hence, in this scenario the weight update equation becomes simply

$$\mathbf{w}_{k}^{(i)} \propto \mathbf{p}(\mathbf{z}_k|\mathbf{x}_k^{(i)}). \tag{2.71}$$

The utility associated with even generic particle filtering algorithms should be evident. Often times, a need arises where the particle representation of a probability density function is inadequate for the task at hand. In situations where particle collections simply will not do, one can consider alternative probability representations. A few of the more popular methods for representing probability density functions from a collection of samples are presented next.

2.5.6 Converting Particles to Probabilities. Given that a particle filter represents a given probability density function with a collection of weights and particles, there is often a need to obtain a more compact representation, as is the case with most multi-agent data fusion processes. At the core of alternate representations for probability density functions is the ability to generate samples from an arbitrary density [88]. More often than not, the structure of probability density functions that are typically dealt with in the fields of guidance, navigation, control, tracking, etc. are not “nice" densities, in that they are typically propagated with nonlinear dynamics models, posses multiple modes, and almost certainly not reasonably described with Gaussian statistics. A compact and efficient representation for these types of densities would be a valuable tool, particularly in the multiple agent localization scenario consider later in Chapter IV. There exists numerous techniques for obtaining compact representations for particle collections in the relevant literature. A few of the more popular techniques are briefly discussed next.

2.5.6.1 Histograms. The histogram is possibly the *easiest* nonparametric density estimator to realize. The only descriptions required in the construction of a histogram are the location for the center bin x_0, and the width of each of the bins h (assuming variable bin widths are not allowed). With the bin centers and the bin width one can define the actual bins in the histogram via

$$I_n = [x_0 + nh, x_0 + (n+1)h]\,, \tag{2.72}$$

where $n = [..., -1, 0, 1, ...]$. Once the bins have been defined, the histogram simply becomes the number of samples falling within a bin divided by the total number of samples times the bin width which can be expressed mathematically according to

$$\hat{H} = \frac{I_i \times x_m}{k\Delta h}, \tag{2.73}$$

where I_i denotes a particular bin, x_m denotes the number of samples within bin I_i, k is the total number of samples, and h is the bin width. Caution is needed when considering using a histogram for the representation of a probability density. The main concern is that there is a possibility that the density representation will become discontinuous. Discontinuities will occur in the event that the number of samples in a particular bin is 0. In an attempt to overcome this issues, alternative techniques have been developed. For example, an alternative representation method of particular interest in this research is known as the Gaussian Mixture Model (GMM).

2.5.6.2 Gaussian Mixture Models. The goal of GMM is to take a collection of samples that have been drawn from an arbitrary density and to select the *“best"* mixture components (Gaussian densities) that represents the data. The Gaussian mixture is calculated in the following fashion

$$\mathbf{p}(\mathbf{x} \mid \theta) = \sum_{k=1}^{K} \omega_k \mathcal{N}(\mathbf{x} \mid \mu_k, P_k), \tag{2.74}$$

where the parameter vector is $\theta = \{\omega_k, \mu_k, P_k\}$, and carries the values that define each component of the mixture. The ω_k are weighting terms, and adhere to the following normalization constraint

$$\sum_{k=1}^{K} \omega_k = 1, \qquad 0 \leqslant \omega_k \leqslant 1. \tag{2.75}$$

Now, in order to determine what components should be selected, generally requires solving a constrained optimization problem. The constraints are given in Equation (2.75). There is an additional constraint that imposes symmetry and positive semi-definiteness on each component's covariance matrix. Gradient-descent type algorithms can be used to try and solve for the GMM parameters [151]. However, a closed form solution to the optimization problem generally does not exist [38].

An alternative technique to gradient-based optimization algorithms is the Expectation-Maximization (EM) algorithm. The EM algorithm starts with an initial guess of the parameters for the GMM and iterates back and forth between the two steps outlined next. The following algorithm presentation can be found in the influential book of Bishop [38].

2.5.6.3 Expectation Maximization Algorithm. The EM algorithm for Gaussian mixtures is defined according to the following steps: [151]

- **Expectation Step:** Let the current parameter vector θ contain the current parameters for each component of a GMM. The first step is to compute the mixture weights according to

  $$\frac{\omega_k \mathbf{p}_k(\mathbf{x}^i \mid \theta_k)}{\displaystyle\sum_{m=1}^{K} \omega_m \mathbf{p}_m(\mathbf{x}^i \mid \theta_m)}, \qquad \text{where} \qquad 1 \leqslant k \leqslant K \quad \text{and} \quad 1 \leqslant i \leqslant N, \tag{2.76}$$

  for all data points $\mathbf{x}^i$ and mixture components k. Notice that for each data point, the weights are defined such that they sum to one. This results in a matrix of size $N \times K$ of weights with each row summing to one.

- **Maximization Step:** The newly calculated weights and available data are used to calculate new parameters for the GMM. This is done via

$$\mu_k = \frac{\sum_{i=1}^{N} \omega_k \mathbf{x}^i}{\sum_{i=1}^{N} \omega_k}. \tag{2.77}$$

  The new mean is calculated in much the same way that a standard empirical average is computed, with the exception that the i^{th} data point has a fractional weight. The updated covariance is found according to

$$P_k = \frac{\sum_{i=1}^{N} \omega_{i,k} (\mathbf{x}^i - \mu_k)(\mathbf{x}^i - \mu_k)^T}{\sum_{i=1}^{N} \omega_{i,k}}. \tag{2.78}$$

A representation of the components of a Gaussian mixture can be seen in Figure 2.18.

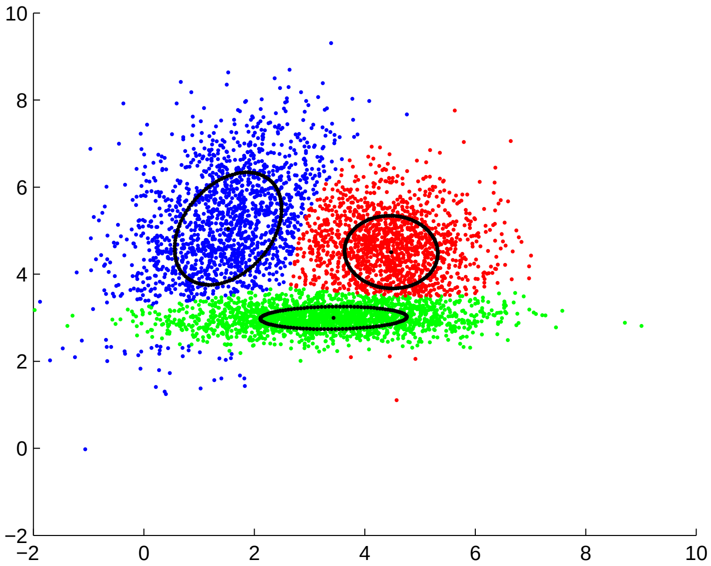


Figure 2.18: A Gaussian mixture model implementation with the Expectation Maximization algorithm. The scenario considered utilized 5000 samples.

2.5.6.4 Kernel Density Estimation. It is a widely known fact that a GMM can represent any density to a level of precision commensurate with the number of components in the mixture. What if, as suggested by [88], instead of trying to find a minimum number of appropriate Gaussian components in a mixture, a Gaussian kernel is simply assigned to every data point. The process just described is known as kernel density estimation (KDE) or Parzen window estimation. A KDE is a nonparametric density estimation technique, much like the histogram representation. Nonparametric here refers to the fact that the number of parameters increases linearly with respect to the number of data samples, and not that the density doesn't have parameters [38], [151]. Generally speaking, a KDE model is represented by

$$p(\mathbf{x} \mid h) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{h^d} \mathcal{K} \left(\frac{\mathbf{x} - x_n}{h} \right). \tag{2.79}$$

The kernel function $\mathcal{K}$ is associated with a parameter h which is called the bandwidth, similar to the case for the histogram estimator. If, for example, the kernel was given by

$$\mathcal{K}(u) = \mathbf{exp} \left\{ \frac{-\|u\|^2}{2} \right\}, \tag{2.80}$$

then each data sample can be considered a component of a GMM with a mean of $\mu_n = \mathbf{x}_n$ and variance $P_n = h^2 I_d$ [38]. So

$$\|P_n\|^{\frac{-1}{2}} = \frac{1}{h^d}, \tag{2.81}$$

and

$$\omega_n = \frac{1}{N}. \tag{2.82}$$

In general, any kernel function can be used provided that it is positive semi-definite

$$\mathcal{K}(u) \geqslant 0, \tag{2.83}$$

and

$$\int \mathcal{K}(u)du = 1. \tag{2.84}$$

Finally, the parameter h is also called a smoothing parameter, and it does just that. It determines how smooth the resulting density estimate is. The ability of kernel density implementations to produce accurate estimates is intimately tied to the smoothing parameter h, as was the case in the histogram estimator. The result of a kernel density estimation algorithm using Gaussian densities with unit variance can be seen in figure 2.19.

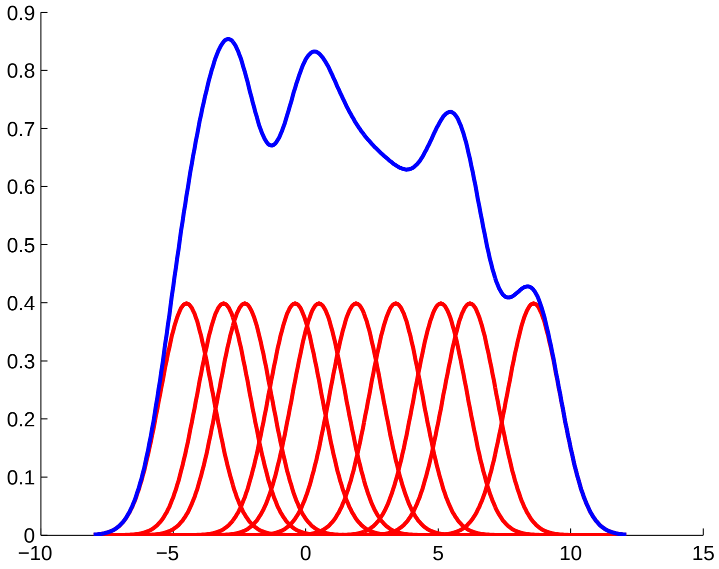


Figure 2.19: Kernel density estimation with Gaussian kernels. The scenario considered utilized 10 kernels with all having unit variance and mean values given by $[-4.5, -3.1, -2.3, -0.4, 0.5, 1.9, 3.4, 5.1, 6.2, 8.6]$.

2.5.7 Discussion. The generic particle filtering algorithm is quite a powerful filtering tool, because it imposes no restrictions on the severity of the model nonlinearities nor on the family of noise distributions that are acceptable. Unlike the EKF algorithm, the models do not need to be analytic *i.e.,* they can be discontinuous. Furthermore, the particle filter is ideally suited for dealing with densities that are faithfully described as being multi-modal in nature, a scenario that the EKF and UKF simply can not address.

The generic algorithm for a particle filter is a rather simple algorithm. Moreover, it is capable of obtaining superior estimation results for several nonlinear and non-Gaussian estimation problems. In cases where parametric filters such as the EKF and UKF are inadequate, the generic particle filter can be used for an estimation procedure with minimal effort from a user. However, as stated by Daum [98], this is actually a mixed blessing. The fear expressed by Daum is that challenging scenarios will not be treated with the appropriate level of respect that they deserve, and that nuances of a particular problem definition will not be appreciated as they should.

2.6 Decentralized Particle Filtering

Decentralized filtering can be found in the literature as early as 1979, with the influential work by Speyer [263]. Recent decentralized nonlinear filtering formulations have benefited from the original work of Speyer. For example, modern methods based on consensus filtering approaches [209], [232], and diffusion process strategies [176] have roots that can be traced back to the work of Speyer. Some recent surveys identify portions of the vast range of available algorithms from as equally vast research disciplines are available in [239], [44], [4], [200], and [292].

In systems described as having ad-hoc communication networks, a need for addressing the fusion of dependent data arises. Dependency can occur in a few ways. One way is through the use of common process models used to describe the temporal evolution of states [250]. Another reason for the existence of dependent data is the common measurement history that is manufactured when agents exchange data repeatedly [250]. For the previously mentioned reasons, to assume that state estimates generated among multiple agents are independent is generally a bad assumption in practice [250]. In fact, the only ways to ensure that a fused estimate is the result of truly independent pieces of data is to maintain a database of all of the communicated data among all of the agents for the entire mission horizon, or to place overly restrictive constraints on the communications topology used among agents [136], [110].

Motivation for considering the use of particle filters was just presented in the previous section. Particle filters were shown to enjoy key attributes not afforded to the more *parametric* approaches to Bayesian estimation like extended and unscented Kalman filters. Some of the more notable advantages include the ability to represent arbitrary probability density functions, the removal of model noise statistical requirements, and no longer needing to linearize estimation models. However, particle filters do present challenging dilemmas when considered for use in decentralized data fusion scenarios.

Two fundamental issues arise with the fusion of particle collections. The first issue is that for any two collection of particles, there is no guarantee that the support for one collection will mirror the other. Likewise, there is no guarantee that any particle in either collection will be collocated [169], that is

$$\operatorname{supp}\left(\left\{\mathbf{x}_A^i\right\}\right) \neq \operatorname{supp}\left(\left\{\mathbf{x}_B^j\right\}\right), \qquad \text{where} \qquad i = j. \tag{2.85}$$

Hence, naive fusion of particles is an ill-defined problem as pointed out by Ong *et al.*, [170]. To demonstrate the problem with the naive fusion of particles, refer to Figure 2.20. In Figure 2.20 there are two sets of particles, labeled (A) and (B), that are not collocated. If the two sets of particles were naively multiplied together, then the result would be the empty set shown in the sub-figure labeled (Result).

Another challenge with decentralized particle filtering is how to address the exponential growth in the number of particles associated with the linear increase in state dimensions. The explosion of required particles is a result of the so-called curse-of-dimensionality [275], [223]. In order to fully appreciate the value added and challenges presented by employing a decentralized particle filtering strategy, the standard Bayesian data fusion model is required.

2.6.1 Key Components of Fusion Equation. The following example can be found in the recent handbook authored by Liggins *et al.*, [172]. In the following discussion it is assumed that two agents, A and B, are able to communicate with one another. The

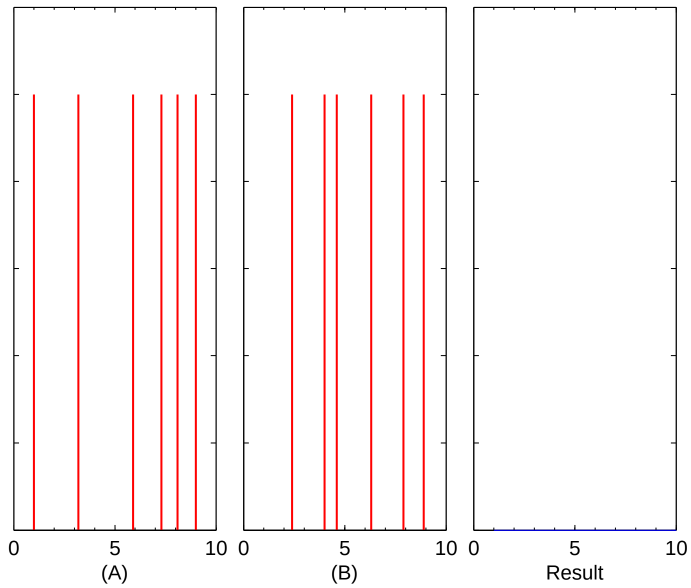


Figure 2.20: Multiplication of two particle collections not collocated (inspired by [167]).

information available to each agent at any given time is comprised of data obtained locally denoted by $\{\mathbf{z}_a\}$ and $\{\mathbf{z}_b\}$ for agents A and B, respectively. Also available to each agent is the common information resulting from repeated communications between agents A and B denoted as $\{\mathbf{z}_c\}$. The combined sets are defined according to

$$\mathbf{Z}_A = \{\mathbf{z}_a, \mathbf{z}_c\} \tag{2.86}$$

$$\mathbf{Z}_B = \{\mathbf{z}_b, \mathbf{z}_c\}, \tag{2.87}$$

where

$$\mathbf{z}_c = \mathbf{Z}_A \cap \mathbf{Z}_B. \tag{2.88}$$

Furthermore, it is assumed that the individual measurements obtained by each agent are conditionally independent of the true state $\mathbf{x}_t$ such that

$$\mathrm{p}(\mathbf{Z}_A, \mathbf{Z}_B \mid \mathbf{x}_t) = \prod_i^N \mathrm{p}(\mathbf{Z}_i \mid \mathbf{x}_t). \tag{2.89}$$

The conditional independence assumption is valid provided that the measurement errors are independent between the sensors used by agent A and B, respectively, in addition to being independent over time [173]. The conditional independence assumption stated in Equation (2.89) leads to the following representation for the shared Bayesian likelihood [173]:

$$\mathbf{p}(\mathbf{Z}_A \cup \mathbf{Z}_B \mid \mathbf{x}) = \frac{\mathbf{p}(\mathbf{Z}_A \mid \mathbf{x})\mathbf{p}(\mathbf{Z}_B \mid \mathbf{x})}{\mathbf{p}(\mathbf{Z}_A \cap \mathbf{Z}_B \mid \mathbf{x})}. \tag{2.90}$$

The diagram shown in Figure 2.21 depicts the types of probability densities shared among agents, and can be found in chapter 17 of [172]. Note, the author's notation for the different probability densities is adopted here. For example $\mathbf{Z}_{A/B}$ denotes the information unique to agent A and not shared by agent B. Under the Bayesian estimation framework used in this dissertation, one needs to determine the probability of the disjoint set $\mathbf{Z}_A \cup \mathbf{Z}_B$ conditioned on the state value $\mathbf{x}$ given by $\mathbf{p}(\mathbf{Z}_A \cup \mathbf{Z}_B \mid \mathbf{x})$. Utilizing Figure 2.21 and dropping the time index k for clarity, the disjoint probability can be obtained according to

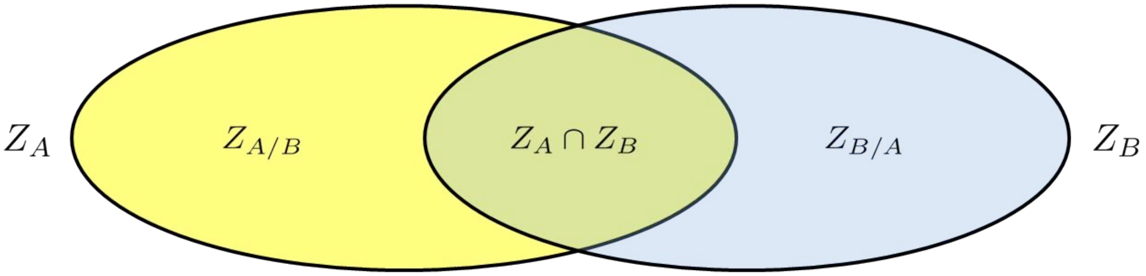


Figure 2.21: Types of data associated with the fusion of dependent data [172].

$$\begin{aligned}
\mathbf{p}(\mathbf{Z}_A \cup \mathbf{Z}_B \mid \mathbf{x}) &= \mathbf{p}(\mathbf{Z}_{A/B} \cup \mathbf{Z}_{B/A} \cup \mathbf{Z}_{A\cap B} \mid \mathbf{x}) && (2.91)\\
&= \mathbf{p}(\mathbf{Z}_{A/B} \mid \mathbf{Z}_{B/A} \cup \mathbf{Z}_{A\cap B}, \mathbf{x})\mathbf{p}(\mathbf{Z}_{B/A} \cup \mathbf{Z}_{A\cap B} \mid \mathbf{x}) && (2.92)\\
&= \mathbf{p}(\mathbf{Z}_{A/B} \mid \mathbf{Z}_B, \mathbf{x})\mathbf{p}(\mathbf{Z}_B \mid \mathbf{x}) && (2.93)\\
&= \frac{\mathbf{p}(\mathbf{Z}_{A/B} \cup \mathbf{Z}_{A\cap B} \mid \mathbf{x})\mathbf{p}(\mathbf{Z}_B \mid \mathbf{x})}{\mathbf{p}(\mathbf{Z}_{A\cap B} \mid \mathbf{x})} && (2.94)\\
&= \frac{\mathbf{p}(\mathbf{Z}_A \mid \mathbf{x})\mathbf{p}(\mathbf{Z}_B \mid \mathbf{x})}{\mathbf{p}(\mathbf{Z}_{A\cap B} \mid \mathbf{x})} && (2.95)\\
&= \frac{\mathbf{p}(\mathbf{Z}_A \mid \mathbf{x})\mathbf{p}(\mathbf{Z}_B \mid \mathbf{x})}{\mathbf{p}(\mathbf{Z}_A \cap \mathbf{Z}_B \mid \mathbf{x})}. && (2.96)
\end{aligned}$$

Equation (2.96) can now be used to obtain the global posterior density $\mathbf{p}(\mathbf{x} \mid \mathbf{Z}_A \cup \mathbf{Z}_B)$ via

$$\begin{aligned}
\mathbf{p}(\mathbf{x} \mid \mathbf{Z}_A \cup \mathbf{Z}_B) &= \frac{\mathbf{p}(\mathbf{X}_A \cup \mathbf{Z}_B \mid \mathbf{x})\mathbf{p}(\mathbf{x})}{\mathbf{p}(\mathbf{Z}_A \cap \mathbf{Z}_B)} && (2.97)\\
&= \frac{\mathbf{p}(\mathbf{Z}_A \mid \mathbf{x})\mathbf{p}(\mathbf{Z}_B \mid \mathbf{x})\mathbf{p}(\mathbf{x})}{\mathbf{p}(\mathbf{Z}_A \cap \mathbf{Z}_B \mid \mathbf{x})\mathbf{p}(\mathbf{Z}_A \cup \mathbf{Z}_B)} && (2.98)\\
&= \frac{\mathbf{p}(\mathbf{x} \mid \mathbf{Z}_A)\mathbf{p}(\mathbf{x} \mid \mathbf{Z}_B)}{\mathbf{p}(\mathbf{x} \mid \mathbf{Z}_A \cap \mathbf{Z}_B)} && (2.99)
\end{aligned}$$

Under the conditionally independent assumption just mentioned, one can also obtain the following representation

$$\mathbf{p}(\mathbf{x} \mid \mathbf{Z}_A \cup \mathbf{Z}_B) \propto \mathbf{p}(\mathbf{Z}_{A/B} \mid \mathbf{x})\mathbf{p}(\mathbf{Z}_{B/A} \mid \mathbf{x})\mathbf{p}(\mathbf{Z}_A \cap \mathbf{Z}_B \mid \mathbf{x})\mathbf{p}(\mathbf{x}), \qquad (2.100)$$

where $\propto$ means proportional to and is needed in the absence of normalization, A and B still denote two agents, the probability density functions $\mathbf{p}(\mathbf{Z}_{A/B} \mid \mathbf{x})$ and $\mathbf{p}(\mathbf{Z}_{B/A} \mid \mathbf{x})$ represent new information, $\mathbf{p}(\mathbf{Z}_A \cap \mathbf{Z}_B \mid \mathbf{x})$ represents the common information shared between the agents, and $\mathbf{p}(\mathbf{x})$ represents the prior probability density. Clearly from Figure 2.21, the common information term resides with both agent A and agent B, and if accounted for yields the following representation of the fusion equation

$$\begin{aligned}
\mathbf{p}(\mathbf{x} \mid \mathbf{Z}_A \cup \mathbf{Z}_B) &\propto \mathbf{p}(\mathbf{Z}_{A/B} \mid \mathbf{x})\mathbf{p}(\mathbf{Z}_{B/A} \mid \mathbf{x})\mathbf{p}(\mathbf{Z}_A \mid \mathbf{x})\mathbf{p}(\mathbf{Z}_B \mid \mathbf{x})\mathbf{p}(\mathbf{x})\\
&\propto \mathbf{p}(\mathbf{Z}_{A/B} \mid \mathbf{x})\mathbf{p}(\mathbf{Z}_{B/A} \mid \mathbf{x})\mathbf{p}(\mathbf{Z}_A \cap \mathbf{Z}_B \mid \mathbf{x})^2\mathbf{p}(\mathbf{x})
\end{aligned} \qquad (2.101)$$

where the incorporation of redundant information is the result of squaring the shared common information among the agents, namely $\mathbf{p}(\mathbf{Z}_A \cap \mathbf{Z}_B \mid \mathbf{x})^2$. The squaring of common information is the reason that the division operation in Equation (2.99) is required.

2.6.2 Interpretation of Fusion Equation. With respect to decentralized filtering, Equation (2.99) has the following interpretation. Observe that the numerator involves the multiplication of local estimates while the denominator is considered to be the common information. The multiplication operation plays the role of incorporating new information from received local estimates, while the division operation takes the role of removing common information between received estimates and locally produced estimates [212], [169], [211], [170], [168]. Clearly, the difficulty in performing decentralized data fusion resides in determining and removing the common information [167] (*i.e.*, the denominator in Equation (2.99)).

There have been several methods suggested in the decentralized data fusion literature. One of the more popular methods is based on the use of information measures. Information interpretations of decentralized data fusion have led to the development of multiple solution approaches, which is why they are the topic of the next section.

2.7 Information Measures

The real question that needs to be asked in reference to divergence measures is, how does one choose from among the seemingly countless published divergence measures? For example, the paper by Sung-Hyuk Cha [60] is a survey of no less than 45 different measures. Additionally, the book by Deza [80] is devoted entirely to definitions of distances, divergences, and similarity measures with various applications for each. There are all types of divergence measures with various degrees of appropriateness for any given problem. Some of the more mainstream measures are presented next.

A rather large number of divergences that appear in the literature belong to a class that was defined independently by Ali and Silvey in 1966 [8] and Imre Csiszar in 1967. This class, in the literature, takes on several different names to include Ali-Silvey-Csiszar Class and f-divergences. Popular examples in this class are the J-divergence, Kullback-Leibler divergence, χ^2-divergence, and Hellinger distance. Another popular distance measure is Bhattacharyya's distance; however, it does not formally belong to this class but does share some similar properties [70], [219]. The sheer volume of material (*eg.*, [34,36,60,80,174]) is not only difficult to understand at times, it is also difficult to discern the applicability of some of the measures.

2.7.1 Kullback-Leibler Divergence. One of the most utilized information measure is know as relative entropy, or more commonly known as the Kullback-Leibler divergence. When concerned with probability density functions, the Kullback-Leibler divergence is often used to represents how similar or how *close* two probability densities $\mathbf{p}$ and $\mathbf{q}$ are to one another. Although available in several different forms, the Kullback-Leibler divergence is often defined according to [40]

$$\begin{aligned} \mathrm{D}_{\mathrm{KL}}(\mathbf{p}\|\mathbf{q}) &= \mathrm{E}_{\mathbf{p}}\left[\ln\left(\frac{\mathbf{p}}{\mathbf{q}}\right)\right] && (2.102) \\ &= \sum_i \mathbf{p}_i \ln\left(\frac{\mathbf{p}_i}{\mathbf{q}_i}\right). && (2.103) \end{aligned}$$

The Kullback-Leibler divergence is often times introduced as a distance metric. Albeit is true that the Kullback-Leibler divergence plays the role of a squared distance on the space probability density functions, it is not a distance in the rigorous mathematical sense. For example, it is not symmetric, that is to say

$$\mathrm{D}_{\mathrm{KL}}(\mathbf{p}\|\mathbf{q}) \neq \mathrm{D}_{\mathrm{KL}}(\mathbf{q}\|\mathbf{p}), \tag{2.104}$$

nor does it abide by the triangle inequality.

2.7.2 Hellinger Distance. As Equation (2.104) states that the Kullback-Leibler divergence is not symmetric, there are times when the metric property of symmetry is desirable. When the property of symmetry is needed, one can make use of the related metric known as the Hellinger distance, defined as

$$\mathrm{D}_{\mathrm{H}}(\mathbf{p}\|\mathbf{q}) = \sqrt{\frac{1}{2}\sum_i \left(\sqrt{\mathbf{p}_i} - \sqrt{\mathbf{q}_i}\right)^2}. \tag{2.105}$$

Frequently, the Hellinger distance is expressed without the leading coefficient of $\frac{1}{2}$. There is no impact to any metric properties due to the omission. The only noticeable difference is in the defining of the support domain. The difference can be seen by the expanding of the valid support domain to the interval $[0, \sqrt{2}]$, in contrast to the interval $[0, 1]$ when included in the definition.

Unlike the Kullback-Leibler divergence, the Hellinger distance is symmetric, and is a true metric. An often valuable relationship that exists between the Hellinger distance and the Kullback-Leibler divergence is the fact that the Hellinger distance lower bounds the non-metric Kullback-Leibler divergence [68]. The implication is that if the Kullback-Leibler divergence converges, then so does the Hellinger distance.

2.7.3 Bhattacharyya Divergence.

The Bhattacharyya divergence is another popular measure of similarity between probability density functions. The Bhattacharyya coefficient [35] between two probability densities is defined according to

$$\mathrm{D}_{\mathrm{BC}}(\mathbf{p}\|\mathbf{q}) = \sum_{i=1}^{N} \sqrt{\mathbf{p}_i \mathbf{q}_i}. \tag{2.106}$$

Geometrically, Equation (2.106) can be interpreted as the cosine of the angle between two n-dimensional vectors

$$[\sqrt{\mathbf{p}_1}, ..., \sqrt{\mathbf{p}_N}] \qquad \text{and} \qquad [\sqrt{\mathbf{q}_1}, ..., \sqrt{\mathbf{q}_N}]. \tag{2.107}$$

If the two probability densities are equal to each other, then the resulting Bhattacharyya coefficient will be

$$\begin{aligned} \cos(\theta) &= \sum_{i=1}^{N} \sqrt{\mathbf{p}_i \mathbf{q}_i} && (2.108) \\ &= \sum_{i=1}^{N} \sqrt{\mathbf{p}_i \mathbf{p}_i} && (2.109) \\ &= \sum_{i=1}^{N} \mathbf{p}_i && (2.110) \\ &= 1, && (2.111) \end{aligned}$$

implying that $\theta = 0$ as expected. Much like the Kullback-Leibler divergence, the Bhattacharyya coefficient is not a true metric either. The authors of [66] proposed the following modification to the Bhattacharyya coefficient such that it does represent a true metric

$$\mathrm{D}_{\mathrm{BH}}(\mathbf{p}\|\mathbf{q}) = \sqrt{1 - \mathrm{D}_{\mathrm{BC}}(\mathbf{p}\|\mathbf{q})}. \tag{2.112}$$

2.7.4 Chernoff Divergence. The Bhattacharyya divergence is a special case of the Chernoff distance, which is defined as

$$\mathrm{D}_{\mathrm{CH}}(\mathbf{p}\|\mathbf{q}) = \max_{0\leqslant\omega\leqslant 1}\left[-\log\left(\int \mathbf{p}(x)^{\omega}\mathbf{q}(x)^{1-\omega}dx\right)\right], \qquad \omega \in [0,1]. \tag{2.113}$$

If one was to arbitrarily define the parameter ω such that $\omega = \frac{1}{2}$, then the result would be the Bhattacharyya distance.

This section presented topics that can be found in the information theory literature. The focus was on measures of information, and the relationships that exist between them. This section effectively ends the presentation of the different topics that will comprise our toolbox in Chapter III. The remainder of this chapter is focused on presenting the most relevant research literature to our effort. The emphasis is on elements of the conservative data fusion literature, differential geometry uses in nonlinear filtering literature, and particle filter realizations in decentralized data fusion literature.

2.8 Detailed Literature Survey of Closely Related Efforts

This section provides a detailed presentation of the literature that is most related to our work. Literature discussing the connections between conservative data fusion methods, differential geometry, and nonlinear estimation with and without particle filters are highlighted. However, before beginning the literature presentation, the precise meaning of a consistent estimate in the present context is required.

2.8.1 What is a Consistent Estimate? Consistent, in the current context, refers to the fact that the fused estimate isn't over confident. Mathematically, consistency can be defined as follows. Consider an estimate of a mean $\hat{\mathbf{x}}$ and the actual or true mean $\mathbf{x}_t$. If the two quantities are differenced, the result is known as the estimation error. Estimation error is denoted by $\tilde{\mathbf{x}}$. Furthermore, associated with the estimation error will be an error covariance denoted by $\mathbf{P}_{xx}$. The process is defined here by

$$\tilde{\mathbf{x}} = \hat{\mathbf{x}} - \mathbf{x}_t, \tag{2.114}$$

and the covariance of the estimation error is

$$\mathbf{P}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}} = \mathbf{E}[\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T]. \tag{2.115}$$

A consistent estimate will be taken to mean the difference between the calculated estimation error covariance in Equation (2.115) and the expectation of the true error covariance resulting in a positive semi-definite matrix *i.e.,*

$$\mathbf{P}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}} - \mathbf{P}_t \succeq \mathbf{0}, \tag{2.116}$$

where the symbol $\succeq$ is used to express the fact that the left hand side of Equation (2.116) represents a positive semi-definite matrix.

2.8.2 Conservative Data Fusion Methods. The concept of decentralized data fusion was introduced back in Chapter I. This section is used to highlight prominent solution techniques currently available. Particular attention is given to the methods that specifically address the need to ensure estimates remain conservative. Conservative in this context refers to not allowing the uncertainty of a fused estimate to be less than the true system uncertainty. Recently, a survey and performance comparison of a few of the more popular methods for conservative data fusion has been published by Chang *et al.,* [150], and is an excellent source of information on the subject matter. Finally, for the purpose of clarity, the reader is advised that throughout this section the terms node, agent, and data source are used interchangeably, and if distinction is necessary it will be explicitly stated.

2.8.2.1 Graphical Approach. An information graph, as defined by Chong *et al.*, [273], is a method of representing the dynamic relationships that develop when the possibility of alterations to available information content is permitted. One can easily imagine situations that could result in changes to available information content. The following so-called information events were highlighted in [273], and later in the works of Liggins *et al.*, [173]

1. When a agent takes a measurement with its own sensors.
2. An observation is received and is used to update an agent's own estimate.
3. A agent communicates its information to other agent in the network.
4. A agent receives information from another agent and uses it to update its own estimate of the environment.

Both Chong and Liggins make use of the common assumption that the measurements are conditionally independent of the state estimate $\mathbf{x}$. The measurements in this context are comprised of all received data at the current time epoch, as well as the measurements generated by sensors housed on a local sensing agent. The information graph then is used to determine the maximum amount of information available to a sensing agent. For another version of a graphical model used for removal of common information see [64].

2.8.2.2 Tree Connection Approach. The concept of fusion trees as presented by Martin *et al.*, [191]. Martin along with his coauthors suggest an approach that is premised on each agent in a network constructing and maintaining it's own information tree. The information tree will be a record composed of the minimal amount of information required to perform fusion calculations. When a tree from one agent is communicated to another agent, the common elements or branches in the tree are identified through a tree search algorithm. The common branches will then be pruned such that only dissimilar branches remain for the combination process.

2.8.2.3 Channel Filter Approach. The authors of [111] and [113] state that the channel filter is used to identify and maintain estimates of common information passed between any two network agents. Equation (2.99) shows that a division operation is required for the removal of common information between agents. The removal of common information should be performed prior to the fusing process. The division requirement, as shown in Equation (2.99), is the primary source of difficulty when considering decentralized data fusion architectures.

Under the Gaussian assumption, the division can be carried out in closed form, effectively removing the common information between two estimates [241]. However, as pointed out by Nettelton [206], the primary reason the channel filter produces consistent estimates is due to the propagation forward to a designated time step of the received information by the channel manager. This operation induces errors into the estimate, effectively inflating the covariance matrix of the received data. The artificially inflated covariance will typically produce a consistent estimate, however a consistent estimate is not guaranteed.

2.8.2.4 Covariance Intersection. If one assumes that two available estimates are independent, then their optimal fusion is performed with the Kalman filter. The optimality guarantee is legitimate only in the case that the filtering models are linear, and when both process and measurement noise statistics are Gaussian. In the case where the estimates become correlated, the update process in the Kalman filter will falsely incorporate the available information content in the estimates multiple times. The immediate result will be an inconsistent estimate where the filter will indicate that an estimate is more certain than the reality of the filtering situation dictates.

An algorithm first presented in [285] known as the Covariance Intersection (CI) algorithm was developed to address the problem of inconsistent estimation. The CI algorithm is an extension of the Kalman update that treats the update as a convex combination of the two initial estimates, given here as

$$\mathbf{P}_c^{-1} = \omega\mathbf{P}_a^{-1} + (1-\omega)\mathbf{P}_b^{-1} \tag{2.117}$$

$$\mathbf{x} = \mathbf{P}_c(\omega\mathbf{P}_a^{-1}\mathbf{x}_a + (1-\omega)\mathbf{P}_b^{-1}\mathbf{x}_b) \tag{2.118}$$

where

$$\sum_{i=1}^{N}\omega_i = 1 \tag{2.119}$$

is chosen based on some heuristic. In [172], the authors suggest selecting ω such that either the determinant of the fused covariance matrix is minimized, or the trace of the fused covariance matrix is minimized.

As pointed out by Hurley [121], to minimize the determinant of the fused covariance matrices has an information theoretic interpretation. Note, if $\{\mathbf{x}_a, \mathbf{P}_a\}$ and $\{\mathbf{x}_b, \mathbf{P}_b\}$ are consistent, then Equations (2.117) and (2.118) will also be consistent for any choice of $\omega \in [0, 1]$, and for any arbitrary level of correlation [136].

The CI algorithm, as given here, represents a simple linear optimization problem which makes it an attractive option when considering timeliness requirements of estimate availability. The shortcomings of the algorithm are that it still hangs its hat on the Gaussian assumption. Also, it can handle only two estimates at a time, which makes it impractical for larger networks. The CI algorithm can be implemented in an iterative process, however the algorithm becomes less stable than a simple batch update according to Farrell [99]. In fact, Franken *et al.,* [103] showed that if the eigenvalues of the two matrices under consideration differ significantly, the nature of the optimization problem becomes quite difficult.

Attempting to address some of the concerns with the Covariance Intersection algorithm, two notable extensions have been developed. The first is known as the Split Covariance Intersection (SCI) algorithm. The SCI was designed to take advantage of the fact that the error in the estimates can be separated into two mutually independent components [172]. The other variant is known as the Bounded Covariance Inflation (BCInf) algorithm [136]. The BCInf algorithm was designed in an attempt to address the increased computational complexity imposed by the SCI algorithm [242] by assuming an upper bound on the absolute value of the cross correlations can be established [136].

2.8.2.5 Covariance Union. The Covariance Union (CU) algorithm [106], [41], [214] can also be used to address the need for ensuring consistent estimation in decentralized networks. However, according to Gardner *et al.,* [106], the original intended usage for the CU algorithm was for database deconfliction. The phrase *database deconfliction* is used to describe the event when spurious or corrupted estimates are introduced to the sensing network. The Covariance Union algorithm has been applied to situations involving ground vehicles [92]. In particular, scenarios involving the need for sensor fusion in automobiles [268] have benefited from the use of the Covariance Union algorithm.

2.8.2.6 Generalized Chernoff Information Fusion. Using the concept of Chernoff information in order to fuse two sets of information demonstrates a natural progression within the framework of decentralized networks; especially given the numerous accounts of the relationship between Chernoff information and data fusion.

To begin discussing the use of Chernoff information for data fusion, one should be made aware that the definition of Chernoff information has two widely used forms. According to Hurley [122], they are

$$D_{Ch}(\mathbf{p}\|\mathbf{q}) = -\min_{0\leqslant\omega\leqslant 1}\left(\ln\left(\sum_{i=1}^{n}\mathbf{p}^{\omega}\mathbf{q}^{(1-\omega)}\right)\right) \tag{2.120}$$

and

$$D^{*} = D_{KL}(\mathbf{p}_{\omega^{*}}(\mathbf{x})\|\mathbf{p}(\mathbf{x})) = D_{KL}(\mathbf{p}_{\omega^{*}}(\mathbf{x})\|\mathbf{q}(\mathbf{x})) \tag{2.121}$$

where $\mathbf{p}_{\omega^*}(\mathbf{x})$ is defined by

$$\mathbf{p}_{\omega^*}(\mathbf{x}) = \frac{\mathbf{p}^{\omega}(\mathbf{x})\mathbf{q}^{(1-\omega)}(\mathbf{x})}{\sum_{i=1}^{n}\mathbf{p}^{\omega}(\mathbf{x}_i)\mathbf{q}^{(1-\omega)}(\mathbf{x}_i)} \tag{2.122}$$

and ω^* is the value of ω such that Equation (2.121) is true. Also, it should be mentioned that D_{KL} is the popular Kullback-Leibler divergence given by

$$D_{KL}(\mathbf{p}(\mathbf{x})\|\mathbf{q}(\mathbf{x})) = \sum_{i=1}^{n}\mathbf{p}(\mathbf{x}_i)\ln\left(\frac{\mathbf{p}(\mathbf{x}_i)}{\mathbf{q}(\mathbf{x}_i)}\right). \tag{2.123}$$

Now from Equation (2.121), the minimization of the Chernoff information can be viewed as selecting the probability density that is equally close, in terms of the Kullback-Leibler divergence, to the two original probability densities [68]. In a similar fashion, one can view the minimization of the Shannon entropy as selecting the probability that is the most informative or produces the largest surprise [121].

The use of the Chernoff information for decentralized data fusion was not without its own shortcomings. For example, the algorithm still required the use of Gaussian densities, but could be extended to accommodate more elaborate densities. The problem was that the extension to other densities is not very intuitive. For this reason, Upcroft et al. [287] looked to extend the work of Hurley to situations where Gaussian mixture models (GMM) would be used for describing atypical probability densities.

In Upcroft's work, the ability to calculate the Chernoff information for a GMM through a crude approximation was developed. Recognizing the need for further improvement, Julier [138] looked to develop a refined approximation. The refinement focused on the use of Chernoff information in conjunction with the CI algorithm so that GMM density estimates could be used for purposes of decentralized data fusion. Although still crude, Julier's algorithm was able to consistently produce superior estimates to those of the algorithm defined by Upcroft and his team. Superior estimates, as used here, implies that a smaller mean squared error was produced. Finally, Julier's algorithm was also shown to yield estimates that were consistent in covariance.

More recently, Farrell et al. [99] looked to further refine previous efforts to use Chernoff information for data fusion. Farrell first noted that the extension of the Chernoff Fusion principle to multiple probability densities could be expressed by

$$\mathbf{p}_{gcf} = \frac{\prod_{i=1}^{n} \mathbf{p}_i^{\omega_i}(\mathbf{x})}{\sum \prod_{i=1}^{n} \mathbf{p}_i^{\omega_i}(\mathbf{x})}, \qquad \text{where} \quad \sum_{i=1}^{n} \omega_i = 1. \tag{2.124}$$

Farrell's algorithm performed on par with existing decentralized data fusion algorithms, validating his approach to decentralized data fusion.

2.8.2.7 Largest Ellipsoid Algorithm. In an attempt to alleviate the need for assumptions on probability density parameterizations for data fusion, the Largest Ellipsoid algorithm (LEA) was proposed by Benaskeur [15]. Concerned about the repeated over inflation of estimated covariance matrices by the Covariance Intersection algorithm, Benaskeur offered the following alternative approach. Not wanting to stray too far from the geometric interpretation of the CI algorithm, mainly that resulting covariance matrix should be premised on the intersection between covariance ellipsoids, Benaskeur attempted to estimate the largest ellipsoid contained within the intersection of covariance ellipsoids. The LEA approach is in contrast to the CI algorithm, in that it doesn't attempt to overestimate the intersection of covariance ellipsoids.

By searching for the largest shared ellipsoid between covariances, Benaskeur was able to repeatedly produce a smaller covariance than the CI approach. Furthermore, the Largest Ellipsoid algorithm provides the added benefit that it can be implemented in networks where access to computational resources are limited.

Recently, Bochardt and Uhlmann have demonstrated that the Largest Ellipsoid algorithm (sometimes referred to as the Minimum Enclosing Ellipsoid (MEE) algorithm), and the Covariance Union Algorithm presented in Section 2.8.2.5 are actually equivalent [214]. The principle concern regarding the Largest Ellipsoid algorithm is limitations that can be imposed through the orientations of perspective covariance ellipsoids which could seriously diminish the ability to determine encapsulated ellipsoids, and in some cases even make it impossible [302].

2.8.2.8 Other Notable Methods. In 2010 the works published by Rendas and Leitao [188] addressed the redundant information problem, which they called the rumor problem, in a novel way. The authors proposed an approach that was based on the concept of Schur dominance. The Schur dominance was used to select the probability density that was the least informative, but was more informative than the probability density functions being considered in the particular fusion step. Also in 2010, Tian *et al.,* [300] proposed a sampling-based Covariance Intersection algorithm, and Blank *et al.,* [252] outlined several alternative methods to the standard Covariance Intersection algorithm.

In the past year, the literature has seen works published that address the redundant information problem in yet still innovative ways outside the traditional methods. For example, Montijano *et al.,* [93] proposed a method based on dynamic voting, Noack *et al.,* proposed an approach based on pseudo-Gaussian probability densities [32], and Reinhardt *et al.,* [187] used set-theoretic methods to address the redundant information problem.

2.8.3 Nonlinear Filtering and Differential Geometry. The proposition of nonlinear estimation algorithms in a differential geometric framework is not a new idea. However, even with published articles that demonstrate the ability to fuse two densities, it appears to be a seldomly considered technique in nonlinear estimation literature. As early as 1990 in a series of publications by Rudolph Kulhavy [155–159], the theoretical work of Rao [236], Efron [95], and Amari [12] began influencing portions of the nonlinear estimation and filtering literature.

The work of Rudolph Kulhavy was primarily concerned with parameter estimation scenarios. In 2003, Zhe Chen authored a technical report [63] where he described the works of Kulhavy. Chen went on to further suggest that the primary contribution of Kulhavy's work was to show that a parameter could easily be approximated by projecting onto a local tangent space. Additionally, Kulhavy proposed the use of conditional inaccuracy as a estimation performance metric. Iltis *et al.*, [124] and later Kulhavy himself extended the tangent space concept to problems where state estimation was the primary focus.

In the work of Beard *et al.*, [234] the nonlinear estimation problem is approached via a slightly different projection technique. The authors utilize a technique known as a Galerkin projection. Essentially, the Galerkin projection is used to approximate the posterior conditional density with a collection of basis functions. Coefficients for the basis functions are what's propagated in time, and ultimately used to estimate the desired probability density.

During the same time period that Kulhavy was publishing his findings, another group was also approaching the nonlinear filtering problem with a differential geometry framework. Francois LeGland and Damiano Brigo [49, 50, 50, 51, 74, 75] were concerned with how the projection of arbitrary probability density functions onto the manifold of Exponential family representations affected the ability to perform nonlinear filtering. In 1996 Brigo *et al.*, [50] first published his work on the projection filter with the completion of his doctoral dissertation. The projection filter was described as a finite dimensional nonlinear filtering technique. Furthermore, the utility of the projection filter in nonlinear estimation problems was made apparent through demonstrations. Demonstration results further solidified the already established synergy between the fields of differential geometry and nonlinear estimation. Brigo, in later efforts [71], showed how to adapt the projection filter to the manifold comprised of stochastic differential equations. Special attention was given to densities belonging to the exponential family of densities.

The projection filter is premised on the orthogonal projection of the Kushner-Stratonovich stochastic differential equation onto the local tangent space. The Kushner-Stratonovich equation governs the evolution of a probability density characterized by a continuous process and continuous measurement update [193], [128], and [50]. Later, the projection was considered by LeGland and Brigo as an attempt to solve the infinite dimensional Fokker-Planck equation (FPE) [77]. This projection was considered with the Fisher information metric associated with the finite dimensional manifold whose elements are exponential probability densities. They conjectured that by projecting the FPE onto a finite dimensional submanifold, that a solution could be obtained without having to try and solve infinite-dimensional integrals.

Several useful properties of the family of exponential densities have facilitated their use in nonlinear filtering applications, but by no means is the exponential family the final answer. For example, if the true density to be estimated is in fact a member of the exponential family then a solution to the estimation problem in this framework is guaranteed to exist and it will be globally unique. However, if the true density is not a member of the exponential family, as is the case in many practical scenarios, the assumption of exponential family membership can lead to undesirable consequences [76]. Additionally, only scenarios that maintain unimodal densities can be considered via the exponential family [171]. The exponential family assumption relieves the common Gaussian assumption often made in nonlinear filtering applications, but it is still fairly restrictive. The restrictions are directly related to the requirement that densities remain unimodal throughout the entire nonlinear filter.

The authors of [71] were able to show two key results. Chief among the results was if simplified exponential families are specified, then the measurement update step in the nonlinear filtering algorithm with discrete time observations is performed exactly (*i.e.,* without error). In later publications, the authors go on to show rigorous proofs of their results [48]. The ability to select the correct exponential family member in a repeatable fashion is no easy task according to Babak Azimi-Sadjadi [21]. Because of the difficulty associated with repeatedly selecting the correct exponential density, Azimi-Sadjadi began considering alternative finite statistical manifolds for estimation.

Azimi-Sadjadi *et al.,* [22–25], were the first to attempt to exploit the nonrestrictive nature of particle filters, in conjunction with the analytical tools of differential geometry. Motivated by the lack of convergence results within the influential works of Francois LeGland and Damiano Brigo, Azimi-Sadjadi and team focused their efforts on obtaining the desired convergence results. Additionally, Azimi-Sadjadi extended previous work based on exponential family assumptions to the manifold comprised of the more general mixture family. The extension to the manifold of mixture family densities was the first attempt to address the approximation of arbitrary multi-modal probability density functions within the geometric formulation of nonlinear filtering.

In the body of work by Azimi-Sadjadi, the emphasis was certainly geared towards theoretical advances. However, the applications that were chosen to demonstrate the theory are of considerable interest to this dissertation. The primary applications considered by Azimi-Sadjadi were navigation-based scenarios. Principle scenarios included the integration of an INS with a GPS receiver, integer ambiguity resolution, and change detection.

In recent years, fusion within the framework comprised of nonlinear estimation and differential geometry has enjoyed utility in a host of additional research communities. For example, one research field that has benefited considerably is computer vision. Tenenbaum *et al.,* [274] uses these techniques to conduct analysis on non-rigid shapes for matching. Kwon *et al.,* [160], [144], [161] conduct particle filtering operations based on the observation that covariance matrices are a member of the Lie Group of symmetric positive definite matrices. The primary applications were based on scenarios involving the need to track environmental features, where propagating and estimating covariance matrices plays a crucial role.

Similar covariance tracking techniques can also be found in the works of Park [220, 221], Lee [164], and Wu [298]. Tyagi *et al.,* [282] actually formulate a linear Gaussian Kalman filter in a purely geometric framework for feature tracking applications. Additionally, several authors [162, 224, 240, 264, 279, 305] extend the nonlinear estimation problem to other manifolds to include Steifel and Grassmann manifolds. Steifel and

Grassmann manifolds are useful surfaces when rotations and angles are under consideration.

Recently a series of publications by Mahendra Mallick have focused on the uses differential geometry to determine the severity of nonlinearities in radar and vision tracking problems [26, 183–185]. Finally, other notable uses of the nonlinear filtering and differential geometric framework include investigations into the utility of calculating the geometric mean of a collection of covariance matrices [199], target tracking on nonlinear manifolds [260], [62], [298], [261], the use of manifolds in conjunction with unscented Kalman filtering applications [306], the use of least squares and manifolds to try and provide a solution to the Simultaneous Localization and Mapping problem [284], image and shape-space analysis [5], signal processing algorithms with applications in classification [189], and general nonlinear estimation [291].

2.8.4 Particle Filtering and Decentralized Data Fusion. The advantages of DDF were given in Section 1.2. Clearly, the ability to maintain robustness to unannounced architecture changes and increased network survivability in the event of a catastrophic failure are of considerable value to several application areas. Additionally, the benefits of particle filtering algorithms over other moment-based algorithms (*i.e.,* the EKF and UKF, discussed in Sections 2.4.3.1 and 2.4.3.2), provide considerable advantages in their own right. The flexibility of not having to make assumptions on the parametric form of the probabilistic model, and the seamless ability to accommodate multi-modal density functions are both key ingredients that contribute to the overall value of estimation schemes in a DDF network. This section is dedicated to presenting the relevant literature with respect to the design and implementation of particle filtering algorithms in distributed and/or decentralized frameworks.

The first instantiations of distributed and/or decentralized particle filter algorithms began showing up in the technical literature in 2003. It is this author's contention that it took until 2003 for the hardware processing capabilities and algorithm development to mature to a point that particle filters could begin to be realistically discussed in this forum. For example, Rosencrantz *et al.,* [244] presented a distributed particle filtering algorithm. Their algorithm addressed communication constraints by selecting a subset of particles that were deemed *most informative*. Also, they proposed a strictly query-response communication protocol in which only neighboring nodes could communicate a subset of particles with one another. There is no doubt that the work of Rosencrantz *et al.,* has been influential in nearly all publications related to decentralized particle filtering since their inaugural publication.

However, the algorithm presented by Rosencrantz *et al.,* [244] does suffer from some very serious drawbacks. Most notably, the choice to only transmit a subset of particles, albeit bandwidth friendly, completely ignores the fact that common information may be present in the particle subsets. The primary influence the algorithm conveyed was the evidence that consistent estimates could be obtained provided that a time history of states was maintained. Maintaining a time history clearly violates the properties of a DDF system. Nevertheless, research efforts relating to the use of particle filters in a distributed or decentralized system architecture was initiated by Rosencrantz.

Also in 2003, Bashi *et al.,* [28] presented work concerning distributed particle filtering. They proposed 3 strategies for distributing the generic particle filtering algorithm in an attempt to achieve *“real-time"* performance. The first, Global Distributed Particle Filter (GDPF) would drawn samples and calculate importance weights locally, and the normalization of importance weights and resampling took place externally at a fusion center. The second, Local Distributed Particle Filter (LDPF) algorithm performed all the above operations locally at each node, and a subset of particles were communicated to the fusion center for integration and updating. The third, Compressed Distributed Particle Filter (CDPF) is a combination of the previous two algorithms. The CDPF algorithm adopted the GDPF procedure, but only communicated a *representative* probability density function to the fusion center. The theme of the work led by Bashi was hardware implementations of particle filters. Since all of the data fusion occurred within a centralized fusion center, the need to account for common information was considered intrinsically.

In 2004, Ihler *et al.*, [123] presented work geared towards calibrating sensors that comprise a network. The main contribution of the research presented by Ihler was a communication protocol algorithm rooted in machine learning theory. Ihler, along with his partners, showed that their communication algorithm was capable of converging to the true density under a multitude of scenarios. Nevertheless, the proposed algorithm did not protect against over confident estimates. In fact, the problem of incestuous incorporation of information was not considered at all.

The year 2004 saw a series of publications from the Australian Research Council (ARC) Centre of Excellence in Autonomous Systems (CAS) at the University of Sydney, Australia. The first in the series, authored by Ridley *et al.*, [241], employed the use of Parzen density estimates [222] to represent a continuous version of an empirical particle filtering density. The authors provide an approximate solution to the decentralized particle filtering problem by noting two key facts. First, the use of a Gaussian kernel in the Parzen density estimate has favorable characteristics over other kernel functions. Second, that the division of a Gaussian density by another can be carried out under certain nonrestrictive regularization conditions [218]. Even though the authors provide a closed form approximation for the fusion of Parzen density estimates, they were faced with the constant growing number of density estimates after an update. The authors simply maintained the N highest weighted samples after an update. The key ramification of the aforementioned pruning strategy is that increased emphasis is given to the peaks of a probability density at the cost of essentially ignoring the tails of the probability density.

Finally, it is worth mentioning the work of Coates [65]. He presented two algorithms for distributing a particle filter. The first was predicated on parametric representations of the density and the factoring of the likelihood. The second introduced the idea of a predictive scalar quantization that was used in conjunction with the Lloyd-Max algorithm

[251] to permit adaptive encoding of measurements. The process of adaptive measurement encoding was considered in an attempt to relieve communication bandwidth constraints.

A flurry of research activity, originating from the University of Australia, can be seen in the literature in 2005 [212], [287], and [169]. Essentially, they compared the utility of using Gaussian Mixture Models (GMM), Parzen density estimation, and pure particle representations for DDF. The conclusion was that the GMM representation was the preferred method for representing particle densities when bandwidth efficiency was a high priority (*i.e.,* it required the fewest components). Beyond that, GMM provided a superior representation of particle densities over representations obtained with Parzen density estimate with the same number of components. The number of required components was determined by evaluating the Bhattacharyya coefficient. If the calculated Bhattacharyya coefficient was 0.95 or greater, then it was determined that the correct number of components has been used. However, if a higher priority was given to the accuracy of fusion results, then the use of Parzen density estimates was preferred. When considering the common information problem, the authors developed an extension to the standard CI algorithm such that GMM components were fused. Even though consistent estimates were obtained, a supporting mathematical proof was not supplied. Moreover, component pruning was conducted off-line using the iterative Expectation-Maximization (EM) algorithm.

Also in 2005, Sheng *et al.,* [254] presented a distributed particle filtering algorithm where they too suggested the conversion of the particles to a GMM representation to reduce communication constraints, but they assumed up front that all of the node's information was uncorrelated. Consequently, the uncorrelated information assumption did away with the need to address the data incest problem.

In 2006, the University of Sydney group produced the following publications by Ong [211] and Upcroft [31]. A discernable difference from their previous efforts was the shifted emphasis towards incorporating visual information into the local particle filter. Visual information, in these articles, consisted primarily of image context and feature identities. The research presented by Ong *et al.,* [211] further distinguished itself by choosing to use the generalized Chernoff Information as a means of producing consistent estimation results. Additional noteworthy research contributions included [120], where the focus was on the synergy between the particle filter and control architecture to facilitate search algorithms. The authors of [301] presented a distributed particle filtering algorithm that emphasized agent clustering and each cluster having its own fusion center, leading to a collection of centralized clusters that would only reconfigure if the target being tracked was outside the maximum range of the cluster fusion center. The authors of [47] were the first, to the knowledge of this author, to explicitly incorporate range and bearing information between nodes in the fusion process. Finally, the authors of [127] implemented a semi-distributed particle filtering algorithm that essentially assumed away the redundant information problem by imposing a fixed communication topology.

In the following year, the work of Hoffman *et al.*, was implemented on a quadrotor aerial vehicle. Consideration of realistic timing models by Vemula *et al.*, [289], was one of the first studies of its kind in the field of decentralized particle filtering. Until the work of Vemula and his collaborators was published, it was commonplace to assume a global clock was available in DDF situations. That being the case, all network activities were effectively synchronized. Conversely, Vemula *et al.*, used the particle filter to estimate states, in addition to estimating each agent's local clock bias. Interestingly enough, the authors surmise that the posterior density representing the states and the n^{th} clock bias could be represented effectively using a Beta distribution parametric model.

The year 2008, Ong *et al.*, [168], added to the growing body of decentralized data fusion literature. The principle focus of the authors was on addressing the challenges resulting from the use of channel filters to remove common information. The key source of difficulty with using channel filters resides with the required division operation to remove common information. The authors approached the problem with an innovative algorithm based on importance sampling. Equally as impressive was their solution framework that extended traditional notions of local fusion by including both locally realized and communicated measurements in the local fusion process. Next, by restricting agent kinematics, Li *et al.,* [180] presented a distributed particle filtering implementation that forced agents to take on either a mover or beacon role. The beacons had GPS measurements available, and the movers used received signal strength indications (RSSI) from the stationary beacons as measurements.

Over the past 18 months, multiple algorithms that incorporate an assortment of solution techniques have been published. For example, algorithms using various solution techniques to include multidimensional scaling [94], Markov-chain Monte Carlo sampling with GMM [56], support vector machines [115], both spatial measurements and temporal attributes [13] have been presented. Beyond that, practical applications such as three dimensional map reconstruction [18], and the design of optimal search algorithms [246] have also been published.

2.9 Summary

The principle focus of this chapter was on the presentation of background material vital to understanding subsequent algorithm development. The first topic introduced was differential geometry. Several definitions were provided along with illustrations. The unit hypersphere was spotlighted because of its well understood geometry, and the key role it plays in future algorithm development in Chapter III.

Several traditional and nontraditional methods of data fusion were discussed, where the use of Bayesian probabilistic methods was singled out as being the method of choice in this dissertation. Particle filter solution techniques were given considerable attention. Highlighting the particle filter presentation was a discussion of key advantages provided by particle filters when considering nonlinear models and/or non-Gaussian noise disturbances. Technical challenges associated with decentralized data fusion architectures were presented in detail. Equally important was the comprehensive presentation of current state of the art solution methods for addressing the challenges associated with the realization of decentralized data fusion algorithms. Immediately following the data fusion presentation was a brief introduction of information measures, with an emphasis on more mainstream measures.

The current chapter concludes with a comprehensive chronological survey of the research literature that is most closely related to the research described in this dissertation. In the next chapter, decentralized Riemannian particle filtering algorithms are developed. The algorithms take full advantage of the well defined Riemannian geometry of the unit hypersphere to the degree that it becomes the primary surface for performing filtering operations.

III. Novel Approaches to Decentralized Particle Filtering

3.1 Chapter Overview

In the previous chapter, fundamental concepts from differential geometry, Bayesian estimation, and information theory were presented. This chapter will utilize the material presented in Chapter II to address the decentralized particle filtering problem. In particular, we leverage the pioneering works of Rao [236], Čencov [283], Amari [10], among others to develop two novel algorithms for performing decentralized particle filtering. Throughout this chapter, the synergy between the differential geometry used in the defining of statistical manifolds and the usual Bayesian estimation framework will be exploited, resulting in the presentation of the two aforementioned novel algorithms.

In an attempt to demonstrate connections between existing approaches with the proposed methods, geometric interpretations of existing methods are given. The geometric interpretation of current state-of-the-art approaches provides a natural segway into the defining of the filtering manifold used for algorithm development, along with the filtering tools made available by our choice of manifold. Finally, two geometric decentralized particle filtering algorithms are derived. The first algorithm takes an approach based on the intrinsic geometry of the filtering manifold, while the second uses filtering tools made available through reformulating the decentralized data fusion problem in an alternative information geometric framework.

3.2 The Geometry of Existing Methods

Considering a purely geometric approach to decentralized fusion can possess advantages over more conventional methods. In fact, the most common Bayesian decentralized data fusion approaches can also be considered as geometric approaches. The material comprising the remainder of this section is intended to provide a geometric presentation of currently employed data fusion methods.

Julier *et al.*, [138], [139], [277] makes the observation that the popular Covariance Intersection algorithm can be interpreted as a special case of a more general fusion rule known as Normalized Weighted Geometric Mean (NWGM) defined as

$$\text{NWGM} = \prod_{i=1}^{N} \mathbf{p}_i(\mathbf{x})^{\omega_i}, \tag{3.1}$$

under the following constraints

$$\omega_i \geqslant 0 \qquad \text{and} \qquad \sum_{i=1}^{N} \omega_i = 1. \tag{3.2}$$

In Equation (3.1), all $\mathbf{p}_i$ are considered to be probability density functions. Without loss of generality, the remaining presentation of the NWGM fusion rule will consider two probability density components for simplicity. After enforcing the normalization constraint, Equation (3.1) takes the following form

$$\text{NWGM} = \mathbf{p}_1(\mathbf{x})^{\omega}\mathbf{p}_2(\mathbf{x})^{(1-\omega)}. \tag{3.3}$$

In order to ensure the NWGM in Equation (3.3) is a proper probability density it will need to be normalized as

$$\frac{\mathbf{p}_1(\mathbf{x})^{\omega}\mathbf{p}_2(\mathbf{x})^{(1-\omega)}}{\int_{\mathcal{X}} \mathbf{p}_1(\mathbf{x})^{\omega}\mathbf{p}_2(\mathbf{x})^{(1-\omega)}}, \tag{3.4}$$

where $\mathcal{X}$ is the set of all valid $\mathbf{x}$ values. Equation (3.4) has an alternative geometric interpretation. Equation (3.4) is also the definition for the geodesic that connects probability density functions $\mathbf{p}_1$ and $\mathbf{p}_2$ on a manifold [135], [134]. Another relationship immediately apparent from Equation (3.4) can be seen in the denominator by identifying it as the argument to be maximized in the definition of the Chernoff divergence in Equation (2.120). The definition for the Chernoff divergence is stated again for clarity in Equation (3.5), albeit with a discrete representation

$$D_{CH}(\mathbf{p}_1(\mathbf{x})\|\mathbf{p}_2(\mathbf{x})) = \min_{0\leqslant\omega\leqslant 1}\left(-\ln\left(\sum_{i=1}^{n}\mathbf{p}_1(\mathbf{x}_i)^{\omega}\mathbf{p}_2(\mathbf{x}_i)^{(1-\omega)}\right)\right). \tag{3.5}$$

Equation (3.5) possesses some desirable properties. Notice that the argument to be optimized is the logarithm of the convex combination of two probability densities, which is concave with respect to the parameter ω [269]. The negative switches it to a convex function. Concave functions, like convex functions, when optimized guarantee that a global extremal exists and may be unique. In general, existence and uniqueness guarantees are not available in most practical optimization problems.

The relationships between differential geometry, information divergences, and Bayesian estimation theory are extensive. The relationships highlighted in this section are just a few among the several that are currently available in the research literature (*eg.,* see [213], [189], [100], [11], [256], [247]). The intent of this section was not to be exhaustive in the identification of kinships between differential geometry and nonlinear Bayesian estimation. Instead, the goal was to demonstrate that there exists strong theoretical and practical justifications for pursuing a differential geometric solution approach for the problem of decentralized particle filtering.

3.3 Riemannian Structure of Probability Spaces

How one chooses to represent a probability density function will have a significant impact on the available mathematical machinery for determining solutions. The first step in determining a representation for the probability density function is knowing what options are available. Recently, Srivastava *et al.,* [265] has published a paper that demonstrates at least four different representations for probability density functions. The representations presented included a traditional probability density function, a probability distribution function or cumulative distribution function, a log-density function or log-likelihood function, and a square-root density function.

Following the lead of Srivastava *et al.*, the choice is made to adopt the square-root probability density representation. The collection or family of square-root representations of probability density functions is defined as

$$\Psi = \left\{ \psi : [0, T] \to \mathbb{R} \mid \forall t, \psi(t) \geqslant 0, \int_0^T \psi^2(t) dt = 1 \right\}, \tag{3.6}$$

where the limits of integration are chosen, without loss of generality, based on the fact that the family of functions Ψ are required to be non-negative continuous functions. If discrete functions are being considered, the definition for Ψ is adapted such that the integral is replaced with a finite sum. The non-negativity constraint is required to ensure that the functions are unique.

The reason for selecting the square-root density representation can be seen by noting that Equation (3.6) implies that the collection of square-root densities can be regarded as residing on the unit hypersphere [281]. Interestingly enough, another interpretation for Equation (3.6) can be found in quantum mechanics [83]. In quantum mechanics, the above definition for Ψ in Equation (3.6) also represents a collection of functions known as square integrable functions [83]. Furthermore, under the normalization constraint, one obtains the definition for the well-known Schrödinger equation [233], which is used to define the wave amplitude of a particle at a specific time and location [61].

The advantage that the square-root probability density function representation has over others is that the distance metric is assured to exist in both the tangent spaces as well as between the square-root density functions on the surface of the unit hypersphere. The guarantee existence of the distance metric is a direct result of the fact that the unit hypersphere is an embedding in $\mathbb{R}^n$, and hence comes endowed with the usual Euclidean metric. Another benefit for choosing the unit hypersphere embedded in a higher dimensional Hilbert space, is its well understood differential geometric structure, and many of the quantities of interest in our quest are available in known analytical forms.

One might be asking themselves at this point, “Working with square-root densities has a whole lot of benefits, so why don’t I always work with square-root densities?" One reason for not using square-root densities is that, in general, the positive definiteness required by valid covariance matrices is not guaranteed [37], [104]. The existence is directly tied to the choice of distance measure. If the distance associated with geodesics on unit hyperspheres (*i.e.,* great circle) is used, then the positive definiteness of a covariance matrix cannot be guaranteed. However, if instead one chooses the distance metric associated with the chordal distance, then some positive definiteness guarantees become available for covariance matrices.

We have now selected the desired surface that we will perform our calculation for decentralized particle filtering - the unit hypersphere. Given that the solution approach will involve an optimization procedure (gradient descent), the next step is to establish conditions under which we can expect to determine if a solution exists on our choice of surface, and if it is a unique solution or not. Intrinsic tools for statistics on manifolds provide the primary mechanism for establishing existence and uniqueness conditions.

3.4 Intrinsic Statistics on Manifolds

The concepts of an average or the mean of a collection of items is well defined in Euclidean space. It is easily verifiable that if a particular collection of points in $\mathbb{R}^n$ are under consideration, then the usual arithmetic mean

$$\overline{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i, \tag{3.7}$$

will produce the point that minimizes the sum-of-squared distances to the collection of points $\mathbf{x}_i$, in the Euclidean distance sense. The minimization of the sum-of-squared Euclidean distances interpretation of the mean value can be extended to more general spaces as well. To extend the concept requires a reinterpretation of the meaning of distance.

Given that the space that has been selected to work in is not Euclidean, the question now is what meaning does the concept of mean value take on when considering our more general space? More precisely, what is the appropriate meaning of distance needed to define the desired mean value? Some of the more popular interpretations belong to Fréchet [177–179], Karcher [114], Kobayashi [154], Kendall [67], Buss and Fillmore [54], Noakes [207], Oller and Corcuera [210], and Emery [96].

In the works cited belonging to Fréchet [177–179], he was able to determine that the variance defined as

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{n} D(\psi\|\psi_i)^2, \tag{3.8}$$

where $D(\cdot\|\cdot)$ is the geodesic distance, was minimized when the value of ψ was determined to be the mean value. Hence, according to Fréchet, the expectation on a general Riemannian manifold is calculated according to

$$\mu = \operatorname*{argmin}_{\psi\in\mathcal{M}} \mathrm{E}\left[D(\psi\|\psi_i)^2\right]. \tag{3.9}$$

In fact, Fréchet was able to generalize his result to general metric spaces.

As powerful as the results produced by Fréchet are, they offer some considerable challenges when working on non-Euclidean curved spaces. For example, the process of obtaining the Fréchet mean involves solving an optimization problem that involves the geodesic distance function. Generally speaking, the ability to show that the geodesic distance function exists on a particular space is a difficult task. Even if you can prove that it exists, then proving that the result of the optimization problem is unique is just as arduous.

In an attempt to address some of these concerns, Karcher [114] decided to go a different route. Noting that the Fréchet mean is a global mean, Karcher proposed that instead of determining a global minimum of the variance function in Equation (3.8), one should actually be concerned with determining a local minimum, of which the Fréchet variance is a special case.

The Karcher mean is defined as the element $\psi \in \mathcal{M}$, where $\mathcal{M}$ is a particular manifold, that produces a local minimum of Equation (3.8). Using the insight provided by working in a localized region, Karcher was able to define conditions on the manifold and on the probability densities that ensured not only the existence of the mean, but also its uniqueness. Stated here without proof (which is available in [114]), the conditions can be summarized as:

> ”If the support of probability density $\mathbf{p}$ is defined in a regular geodesic ball of radius r *i.e.*, $\Omega \subset \mathbf{B}(\mathbf{p}, r)$ and regular taken to mean that the radius is such that $2r\sqrt{\kappa} < \pi$ with κ representing the maximum curvature of the manifold, then $\varphi(\mathbf{p})$ is a convex function of $\mathbf{p}$, and as such has a unique critical point defined to be the Karcher mean."

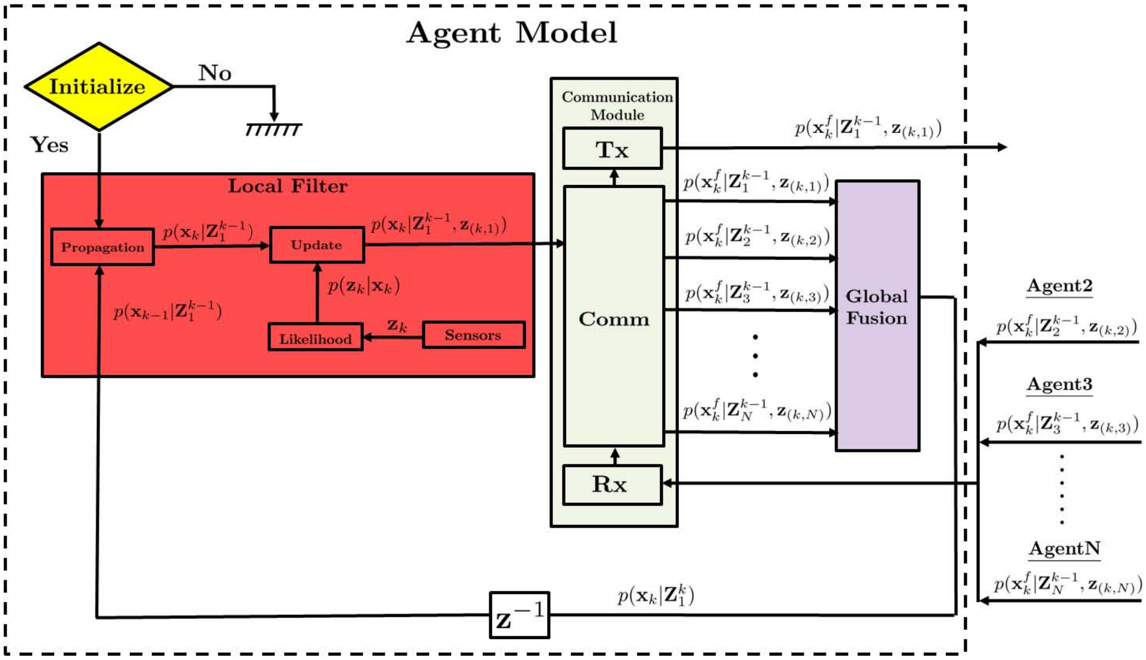
Building on the work of Karcher, Kendall [67] refined the existence and uniqueness conditions established by Karcher. In the case of the unit hypersphere, Kendall showed that the regular ball of radius r was defined such that $r = \frac{\pi}{2}$, which means that as long as the support of $\mathbf{p}$ resided in a open hemisphere, then the Karcher mean will exist, and furthermore it will be unique. One final refinement was contributed by Buss and Fillmore [54] who were able to show that the open constraint on the hemisphere could be relaxed to say that the hemisphere could be closed as long as at least one point of the support Ω was contained in the hemisphere [145].

The method presented for calculating mean values on a Riemannian manifold $\mathcal{M}$ is often times referred to as an intrinsic method, meaning that the actual structure of the manifold and the Riemannian metric were used to determine the relevant statistics. Other approaches for calculating statistics are known as extrinsic. Extrinsic methods are mentioned for completeness purposes only and do not play a role in the work that follows. For its lack of use, the extrinsic mean will not be discussed any further.

The significance of this section was the definition of conditions under which one would be guaranteed the existence of a solution and the uniqueness of the solution in the form of the Karcher mean. However, the existence and uniqueness are with respect to a local minimum. Armed with the knowledge that our iterative procedure, using gradient descent methods, will converge to a unique result (locally), we are now ready to begin putting the pieces together for our approach to decentralized particle filtering.

3.5 Decentralized Riemannian Particle Filter

The term agent, henceforth, is used to denote a mobile platform equipped with local sensors and onboard processing capability. Sensing agents within ad-hoc communication topologies can often be described generically according to Figure 3.1. The generic description in Figure 3.1 will serve as a reference point for algorithm design. In particular, the primary emphasis of the development that follows will be on the sub-processes located in the *Global Fusion* block of Figure 3.1. The processes are divided into two main parts. The first part deals with the decentralized particle filter fusion of global and local data with the use of the differential geometry of the unit hypersphere. The second part deals with the process of updating particle weights based on the results of the decentralized particle filtering approaches presented in the first part.



Agent Model
Initialize
No
Yes
Local Filter
Propagation
$p(\mathbf{x}_k|\mathbf{Z}_1^{k-1})$
Update
$p(\mathbf{x}_k|\mathbf{Z}_1^{k-1}, \mathbf{z}_{(k,1)})$
$p(\mathbf{z}_k|\mathbf{x}_k)$
$p(\mathbf{x}_{k-1}|\mathbf{Z}_1^{k-1})$
Likelihood
$\mathbf{z}_k$
Sensors
Communication Module
Tx
Comm
Rx
$p(\mathbf{x}_k^f|\mathbf{Z}_1^{k-1}, \mathbf{z}_{(k,1)})$
$p(\mathbf{x}_k^f|\mathbf{Z}_1^{k-1}, \mathbf{z}_{(k,1)})$
$p(\mathbf{x}_k^f|\mathbf{Z}_2^{k-1}, \mathbf{z}_{(k,2)})$
$p(\mathbf{x}_k^f|\mathbf{Z}_3^{k-1}, \mathbf{z}_{(k,3)})$
$p(\mathbf{x}_k^f|\mathbf{Z}_N^{k-1}, \mathbf{z}_{(k,N)})$
Global Fusion
Agent2
$p(\mathbf{x}_k^f|\mathbf{Z}_2^{k-1}, \mathbf{z}_{(k,2)})$
Agent3
$p(\mathbf{x}_k^f|\mathbf{Z}_3^{k-1}, \mathbf{z}_{(k,3)})$
AgentN
$p(\mathbf{x}_k^f|\mathbf{Z}_N^{k-1}, \mathbf{z}_{(k,N)})$
$\mathbf{z}^{-1}$
$p(\mathbf{x}_k|\mathbf{Z}_1^k)$

Figure 3.1: Generic diagram of a sensing agent

Before proceeding, a few words regarding the naming conventions used in Figure 3.1 are in order. The use of the bold face capital $(\mathbf{Z})$ is used to denote the collection of available measurements, to include both measurements derived locally and received estimates of landmark states from neighboring agents. The subscript on $(\mathbf{Z})$ is used to identify which particular agent's measurement collection is being referenced. As is standard, the use of the lower case (k) is for identifying particular discrete time steps, and the bold face $(\mathbf{x})$ is used to denote the state vector of the appropriate dimensions for the respective agent. The state vector $(\mathbf{x})$ adopts the agent identification given to the collection of measurements on which it is conditioned.

The bold lower case $(\mathbf{z})$ is used to identify measurements that are obtained by onboard sensors only. One might also notice that some of the local measurements $(\mathbf{z})$ are given two subscripts. The subscript values are used to, again, denote a discrete time step (k), along with the numeric value used to identify the agent that has produced the local measurements.

Finally, notice that the probability density functions that are inputs to the global fusion process consist of all available posterior densities resulting from each respective single local filtering cycle. Associated with the posterior densities used in the global fusion process is a superscript (f). The superscript (f) is used to signify that the densities are with respect to landmark states only, and do not represent agent specific states.

3.5.1 Global Update Module.

When filtering on the unit hypersphere, there are three primary tasks required, and they are (1) conversion of the particles into a more useful probabilistic form (histograms in our case), (2) the actual fusion of the available particle representations, and (3) the updating of the local agent's statistics/particles. All three steps are shown in Figure 3.2. At the core of the first fusion process is an optimization algorithm based on classical gradient descent methods, first proposed by Pennec [226] for use in medical imaging analysis.

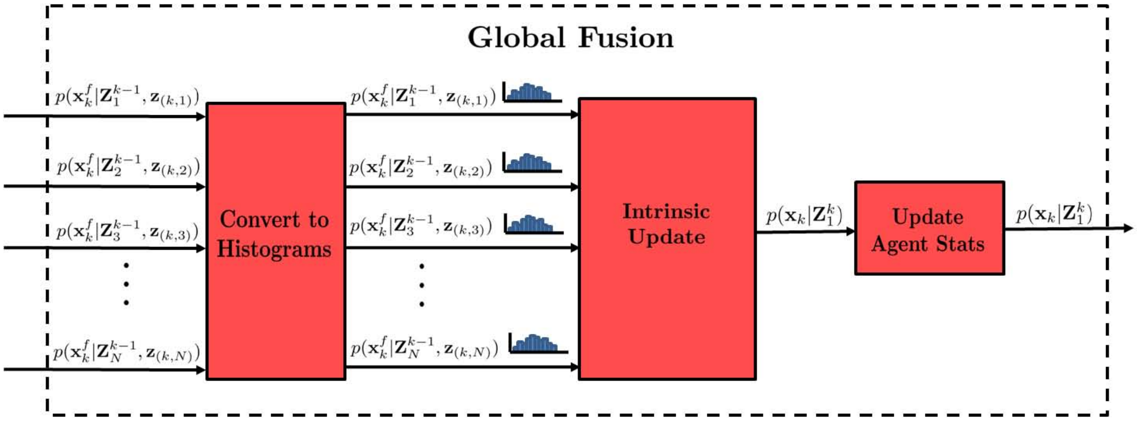

Global Fusion
$p(\mathbf{x}_k^f|\mathbf{Z}_1^{k-1}, \mathbf{z}_{(k,1)})$
$p(\mathbf{x}_k^f|\mathbf{Z}_2^{k-1}, \mathbf{z}_{(k,2)})$
$p(\mathbf{x}_k^f|\mathbf{Z}_3^{k-1}, \mathbf{z}_{(k,3)})$
$p(\mathbf{x}_k^f|\mathbf{Z}_N^{k-1}, \mathbf{z}_{(k,N)})$
Convert to Histograms
$p(\mathbf{x}_k^f|\mathbf{Z}_1^{k-1}, \mathbf{z}_{(k,1)})$
$p(\mathbf{x}_k^f|\mathbf{Z}_2^{k-1}, \mathbf{z}_{(k,2)})$
$p(\mathbf{x}_k^f|\mathbf{Z}_3^{k-1}, \mathbf{z}_{(k,3)})$
$p(\mathbf{x}_k^f|\mathbf{Z}_N^{k-1}, \mathbf{z}_{(k,N)})$
Intrinsic Update
$p(\mathbf{x}_k|\mathbf{Z}_1^k)$
Update Agent Stats
$p(\mathbf{x}_k|\mathbf{Z}_1^k)$

Figure 3.2: Block diagram of the algorithmic steps required during the global update.

3.5.2 Conceptual Preview. The proposed algorithm draws upon several different technologies, each of which have their own standards and traditions. This section is an attempt to ease the reader into key components of the algorithmic process, while avoiding traditional detailed descriptions. Make no mistake, the details are vital and in no way is this section intending to belittle their significance. The more rigorous presentation of key mathematical concepts will follow the conceptual preview of this section in Section 3.6.

At the most fundamental level, the goal of the *Global Fusion* block in Figure 3.1 is to take the data resulting from an agent's local filter iteration and the communicated estimates from a collection of neighboring agents, and combine the data in an attempt to gain a clearer (*i.e.,* more certain) representation of the environment than was previously held.

The assimilation of local and global data begins by constructing histograms of all available landmark data. The histograms are then projected onto the unit hypersphere via a square-root mapping function. Analytical properties of the unit hypersphere, along with its well understood differential geometry make it an attractive surface for the development of data fusion algorithms. In fact, all of the necessary instruments for performing decentralized particle filtering are available in an analytical-form, to include exponential maps, logarithmic maps, and geodesics.

Taking advantage of the available geometry, the method for fusion amounts to finding a mean tangent vector by iterating between logarithmic and exponential mapping functions. Recall that the logarithmic mapping is used to define the tangent vector originating at some point ψ_i on the manifold pointing in the direction of another point ψ_j on the same manifold. The result of the successive logarithmic mapping operations will be a collection of tangent vectors that all originate from an initial starting point ψ_1 pointing towards all other points of interest $\psi_{j=2:N}$.

Once all of the tangent vectors are defined, the goal becomes finding the mean tangent vector, *i.e.,* , the unit vector pointing in the mean direction. This is accomplished by simply calculating the arithmetic mean of all the tangent vectors. Upon calculating the mean tangent vector, it is projected back onto the surface of the manifold by way of the exponential mapping. The result of the final exponential mapping operation will be a new mean square-root density ψ_μ. If desired, the covariance can be calculated as well. Finally, the newly fused square-root pdf is used to update the local agent's particle collection. This algorithm is a gradient descent algorithm. Similar gradient based approaches on Riemannian manifolds can be found in various research articles [299], [9], [261], [281].

Notionally, the steps to be followed are depicted in Figure 3.3. In Figure 3.3 the point ψ_1 represents the initial mean square-root pdf. The tangent vectors $\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, \boldsymbol{\tau}_3, \boldsymbol{\tau}_4$ all originate at point ψ_1 and are pointing in the direction of their respective ending square-root density denoted by $\psi_{2:5}$. The vector $\boldsymbol{\tau}_\mu$ represents the mean tangent vector, which is then projected back onto the manifold with the exponential map to determine the mean square-root density ψ_μ. The conceptual preview hopefully provided a general sense of the mechanics of the first proposed solution method. The remainder of this section is dedicated to addressing our choice to utilize histograms as a way of probabilistically representing a collection of particles.

3.5.3 From Particles to Probabilities. The first task requiring discussion is the process for converting a collection of particles into a chosen probabilistic form. For the

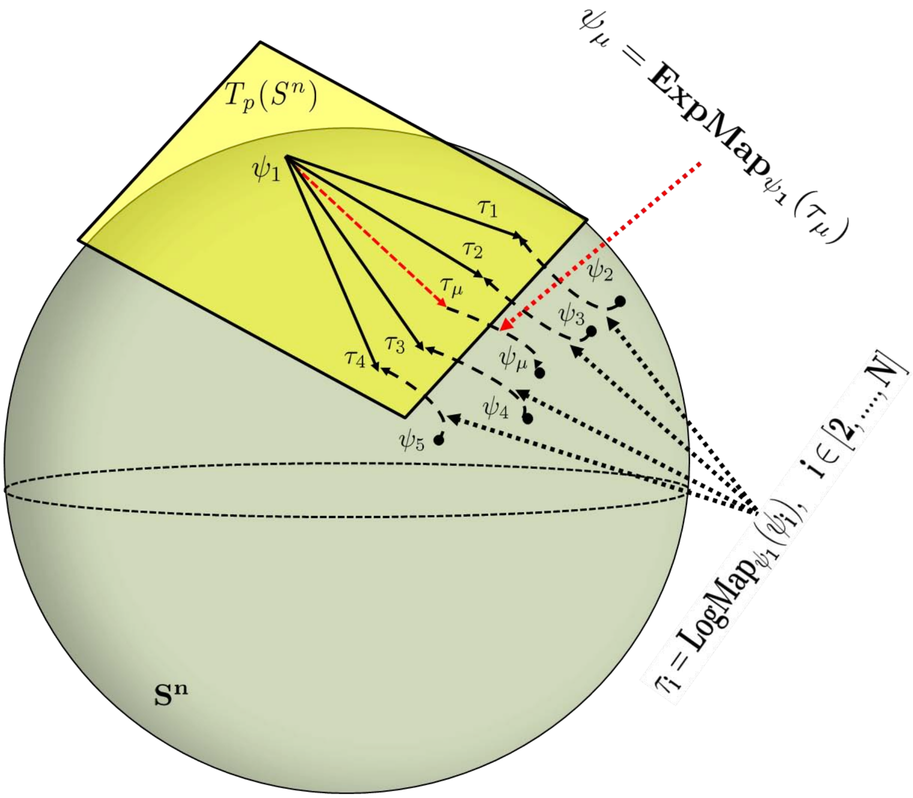

$\psi_\mu = \mathbf{ExpMap}_{\psi_1}(\tau_\mu)$
$T_p(S^n)$
ψ_1
τ_1
τ_2
τ_μ
τ_3
τ_4
ψ_2
ψ_3
ψ_μ
ψ_4
ψ_5
$\tau_i = \mathbf{LogMap}_{\psi_1}(\psi_i),\ \mathbf{i} \in [\mathbf{2},\ldots,\mathbf{N}]$
$\mathbf{S^n}$

Figure 3.3: Conceptual procedure for global fusion.

simulation environment developed as a consequence of this research (detailed in Chapter IV), the histogram was the preferred choice of representation. Histograms are the preferred density representation because of the practical benefits they provide, mainly that only a few parameters are required to be passed between agents in order to reconstruct probability densities. Additional methods for representing a collection of particles with probability densities were briefly described in Section 2.5.6.

A typical histogram produced by a single agent, for a single landmark state, can be seen in Figure 3.4. The histogram in Figure 3.4 was generated with 5000 particles and 100 histogram bins, in addition to being normalized to ensure that the total probability sums to 1.

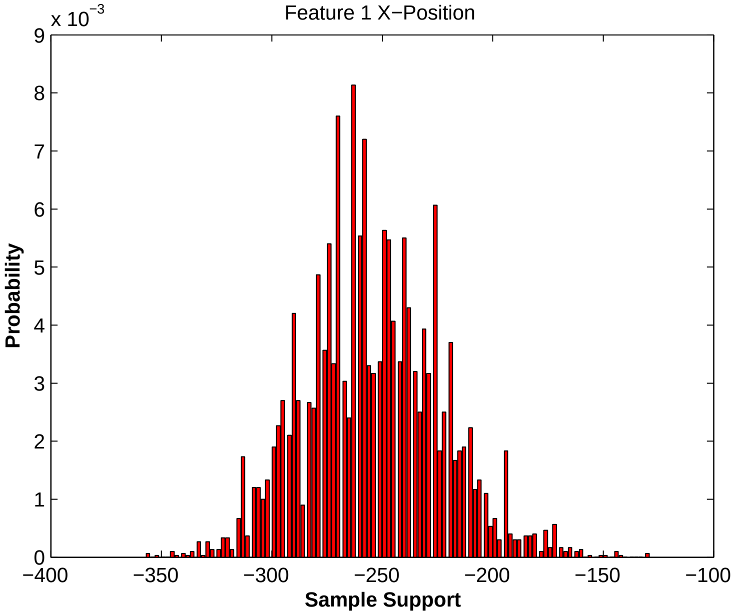


Figure 3.4: Histogram plot of a single dimension landmark state

A reader at this point might be wondering “Why choose a histogram as the way to represent your particles probabilistically?", which is an appropriate question. Sure, there exists techniques capable of producing far superior probability density estimates to those produced by a histogram. However, recall that two of the motivating factors for addressing the decentralized particle filtering problem were to provide an algorithm that is

1) accessible to a broad range of potential users, and 2) computationally efficient. Hence, the choice to use histograms for representing particles probabilistically is justified.

Interestingly enough, for the generic distributed filtering definition given in Equation (2.99), along with the corresponding interpretation given in Section 2.6.2, the use of histograms is not possible. The reason is due to the need to remove common information shared between agents. The removal of the common information was performed with the division operation in Equation (2.99). In the likely event that one or more of the histogram bins will not contain any particles, resulting in a zero probability value for the corresponding support domain, the division operation will become undefined (*i.e.,* can not divide by 0). Alternatively, nowhere in the algorithm presented in Section 3.6 is there a need to perform division.

Another point of discussion concerning our use of the histogram is that we construct our histograms independently for each landmark state. In order to construct histograms based on individual landmark states, one must make the assumption that each landmark state is statistically independent of all of the other states. In reality this is clearly not true. Just consider the fact that the x position and y position states for a single 2D point landmark will share a dependency. The weak assumption of independence is addressed using the same rationale used in the development of the original Covariance Intersection algorithm [132], [285] presented in Section 2.8.2.4. Recall that the existence of an optimal and practical minimum mean-square error fusion algorithm for systems using completely ad hoc communication topologies is not obtainable [288], [258]. Instead, we took inspiration from the original works of Uhlmann, and abandon the pursuit of an optimal algorithm, and instead seek to protect our decentralized particle filtering algorithm from the worst case scenario. The difference can be seen in the analogy that “instead of playing to win, we are playing not to lose". Our first algorithm is described in the next section.

3.6 Algorithm I: Intrinsic Data Fusion Approach

Processes for producing conservative estimates for our first algorithm are shown in Figure 3.5. The first process shown in Figure 3.5 is the projection of probability densities onto the unit hypersphere $\mathcal{S}^n$.

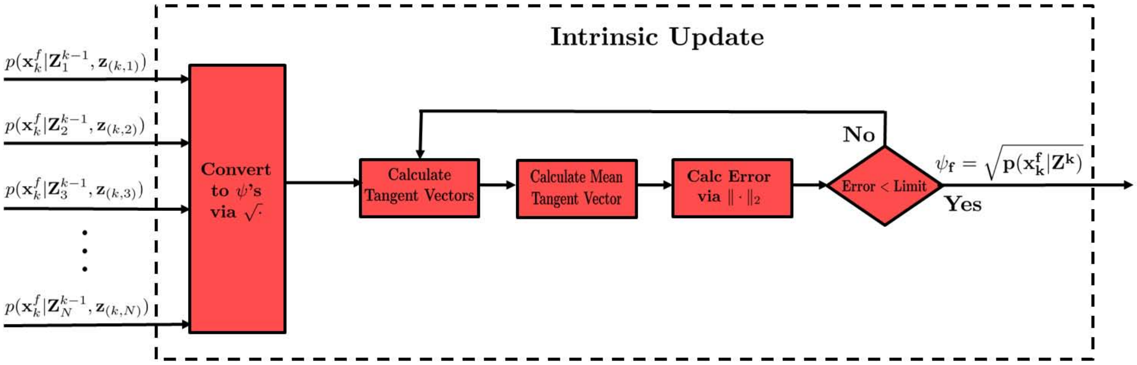

Intrinsic Update
$p(\mathbf{x}_k^f|\mathbf{Z}_1^{k-1}, \mathbf{z}_{(k,1)})$
$p(\mathbf{x}_k^f|\mathbf{Z}_2^{k-1}, \mathbf{z}_{(k,2)})$
$p(\mathbf{x}_k^f|\mathbf{Z}_3^{k-1}, \mathbf{z}_{(k,3)})$
$p(\mathbf{x}_k^f|\mathbf{Z}_N^{k-1}, \mathbf{z}_{(k,N)})$
Convert to ψ's via $\sqrt{\cdot}$
Calculate Tangent Vectors
Calculate Mean Tangent Vector
Calc Error via $\|\cdot\|_2$
Error < Limit
No
Yes
$\psi_\mathbf{f} = \sqrt{\mathbf{p}(\mathbf{x}_\mathbf{k}^\mathbf{f}|\mathbf{Z}^\mathbf{k})}$

Figure 3.5: Decentralized global fusion procedures

3.6.1 Projecting Onto $\mathcal{S}^n$.

The projecting of probability density functions onto $\mathcal{S}^n$ is accomplished in the following manner. First, recall that in order to be classified as a proper discrete probability density function, the following two conditions must be met

$$0 \leqslant \mathbf{p}_i \leqslant 1, \tag{3.10}$$

and

$$\sum_{i=1}^{N} \mathbf{p}_i = 1, \tag{3.11}$$

where $\mathbf{p}_i$ is used to represent sample i from the discrete probability density $\mathbf{p}$. If the conditions in Equations (3.10) and (3.11) are met, then one can define a mapping such that

$$\mathbf{p}_i \mapsto \psi_i = \sqrt{\mathbf{p}_i}. \tag{3.12}$$

It should be mentioned that by virtue of the mapping definition in Equation (3.12), the normalization condition in Equation (3.11) is satisfied via

$$\sum_{i=1}^{N} \psi_i^2 = 1. \tag{3.13}$$

The immediate impact to the present effort is the fact that the collection of square-root densities ψ_i, along with Equation (3.13), guarantees that the magnitude of the vectors represented by $\{\psi_i\}_{i=1}^N$ in Euclidean space $\mathbb{R}^n$ will be exactly one, and hence can be interpreted as a unit vector residing on the surface of the unit hypersphere $\mathcal{S}^n$.

Ultimately, the projection operation amounts to no more than taking the square-root of the probability density function. All available probability densities considered by a single agent for global fusion must reside on the surface of the unit hypersphere before continuing. Next, the calculation of tangent vectors, associated with all available square-root probability densities, is performed.

3.6.2 Tangent Vector Calculation. The calculation of tangent vectors is done with the use of the logarithmic mapping given in Equation (2.18). Recall the logarithmic map takes the geodesic with endpoint ψ_i that originates at point ψ_1 and maps it to the unique vector $\boldsymbol{\tau}$ that is both tangent to point ψ_1 and points in the direction of endpoint ψ_i. Furthermore, the vector $\boldsymbol{\tau}$ will posses the characteristic of constant velocity over the interval defined by the geodesic endpoints.

On the surface of $\mathcal{S}^n$, the logarithmic mapping is defined in a two-step process given previously in Equation (2.20), and is stated here as a matter of convenience, but with ψ_2 replaced with ψ_i for generality

$$\boldsymbol{\tau}_i = \frac{\boldsymbol{\tau}_1 \arccos(\langle \psi_i | \psi_1 \rangle)}{\sqrt{\langle \boldsymbol{\tau}_1 | \boldsymbol{\tau}_1 \rangle}}, \tag{3.14}$$

where

$$\boldsymbol{\tau}_1 = \psi_i - \langle \psi_i | \psi_1 \rangle \psi_1. \tag{3.15}$$

Once all of the necessary tangent vectors are calculated, the task becomes one of finding the mean tangent vector.

3.6.3 Mean Tangent Vector Calculation. When calculating the mean tangent vector $\overline{\boldsymbol{\tau}}$, the result will be a vector pointing in the mean direction *i.e.,*

$$\overline{\boldsymbol{\tau}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\tau}_i, \tag{3.16}$$

where N designates the total number of tangent vectors $\boldsymbol{\tau}$ to be averaged.

3.6.4 Projection Back Onto $\mathcal{S}^n$. The projecting of the mean tangent vector back onto the surface of $\mathcal{S}^n$ makes use of the following properties of geodesics. First, geodesics are the shortest length paths between two distinct points along the surface of a manifold. Second, tangent vectors can be used to uniquely define geodesics. These properties were used to define the exponential mapping operation defined for the manifold $\mathcal{S}^n$ given previously in Section 2.3.2, and again here as

$$\mathrm{ExpMap}_{\psi_1}(\overline{\boldsymbol{\tau}}) = \cos(\|\overline{\boldsymbol{\tau}}\| \cdot \mathbf{t})\psi_1 + \sin(\|\overline{\boldsymbol{\tau}}\| \cdot \mathbf{t})\frac{\overline{\boldsymbol{\tau}}}{\|\overline{\boldsymbol{\tau}}\|}, \tag{3.17}$$

where $\mathbf{t}$ serves as the parameterization, and is constrained such that $\mathbf{t} \in [0, 1]$. Also, to ensure that the exponential mapping in Equation (3.17) is a bijection (*i.e.,* one-to-one and onto), the norm of the mean tangent vector is restricted such that $\|\overline{\boldsymbol{\tau}}\| \in [0, \pi)$.

3.6.5 Calculate Error. Error calculations in this gradient descent algorithm is actually rather simple. The calculation of the error is used for the purpose of designating a stopping criteria for the gradient descent algorithm. At the start of the iteration a minimum acceptable threshold for exiting the iterative algorithm must be defined. Generally, the stopping criteria is defined according to

$$\|\overline{\boldsymbol{\tau}}\|_2 < \mathrm{Threshold}. \tag{3.18}$$

How to define the threshold in Equation (3.18) is the obvious next question. This is where familiarity with the governing physics of the process that is being optimized becomes useful. Essentially, the threshold is determined based on a combination of an acceptable value discerned from what is being optimized, along with what will be utilizing the result of the optimization. Note, given that the gradient descent method is used to calculate a local minimum, caution must be used in selecting the mean initialization value for $\overline{\psi}_0$.

3.6.6 Deciding to Continue or Not. If the algorithm has converged to an acceptable value, the result will be a mean square-root density. The newly selected mean square-root density $\overline{\psi}$, is then used to update the local statistics and particle collection.

At this point, all of the required quantities for updating the local particles have been obtained. The focus now shifts to updating the local particle collection with the results. The algorithmic procedures for updating the local particles with the results of the intrinsic global fusion process can be seen in Figure 3.6.

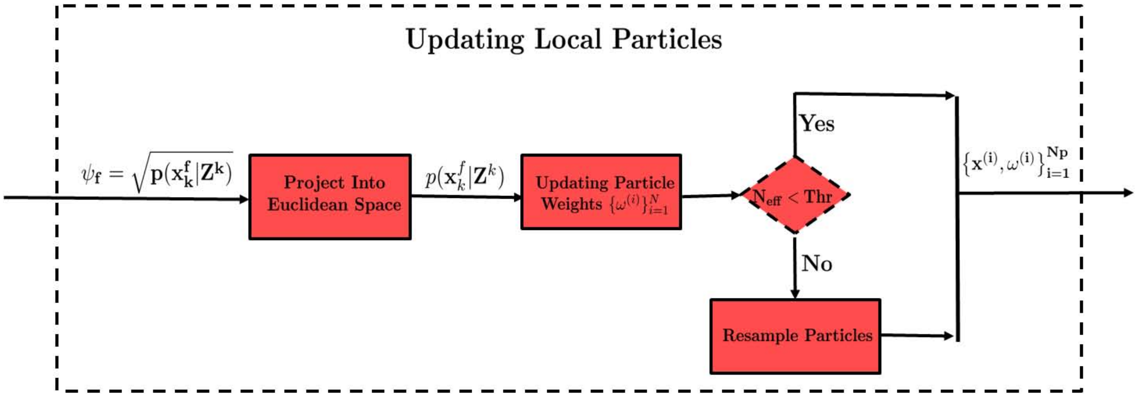

Updating Local Particles
$\psi_f = \sqrt{p(x_k^f|Z^k)}$
Project Into Euclidean Space
$p(x_k^f|Z^k)$
Updating Particle Weights $\{\omega^{(i)}\}_{i=1}^N$
Yes
$N_{eff} < Thr$
No
Resample Particles
$\{x^{(i)}, \omega^{(i)}\}_{i=1}^{Np}$

Figure 3.6: Algorithmic steps required for updating local particles

3.6.7 Returning Back to Where We Started. The projection back into Euclidean space merely amounts to providing the inverse operator to the $\overline{\psi}$ chosen via the gradient descent algorithm just presented. The inverse operator is exactly what one might suspect, the squaring operation which is defined generically according to Equation (3.19)

$$\psi_i \mapsto \mathbf{p}_i = \psi_i^2. \tag{3.19}$$

Once back in Euclidean space, the task becomes one of updating the weights for the collection of particles. Procedures for updating the particle weights will be discussed in Section 3.8.

3.6.8 A Brief Discussion. As previously stated, relationships have been established between the unit hypersphere and traditional Euclidean space long before being considered by this author in this dissertation. Of particular use to the work presented in this dissertation were the established correspondences in the context of Bayesian nonlinear filtering applications. Leveraging the correspondences between the two spaces, we were able to adapt existing technologies from multiple research disciplines into a cohesive algorithm that at its core is a gradient-based optimization procedure. To this author's knowledge, this is the first algorithm designed with the intent of performing decentralized particle filtering on the Riemannian manifold $\mathcal{S}^n$. Further distinguishing characteristics from existing algorithms used in decentralized particle filtering applications include the use of only the intrinsic geometry of $\mathcal{S}^n$ for calculating solutions in conjunction with the use of histograms for representing particles in a probabilistic fashion.

A decentralized particle filter was successfully formulated and implemented using the Riemannian manifold $\mathcal{S}^n$. However, the gradient-based algorithm suffers from a noteworthy limitation. In the algorithm's current presentation, a mechanism for guarding against estimates with an associated covariance that is smaller than reality *i.e.,* overly optimistic is lacking. Motivated by this shortcoming, in Section 3.7, we abandon the gradient-based optimization approach for an alternative method that makes use of the information divergences presented in Section 2.7, along with the differential geometry of the unit hypersphere.

3.7 Algorithm II: Information Geometric Approach

Motivated by the need to protect against overly optimistic data fusion, and the undesirable discarding of an undetermined amount of potentially valuable information, we seek an alternative approach to the optimization-based algorithm outlined in Section 3.6. In spite of the limitations of the gradient-based approach, it has shown that the ability to formulate and implement decentralized data fusion concepts using the differential geometry of the unit hypersphere is not without merit, and that further exploration is justified. Before outlining our alternative formulation, we first establish necessary relationships between components used in our approach with existing decentralized data fusion approaches. The relationships are first discussed at a conceptual level, similar to the presentation in Section 3.5.2.

3.7.1 Establishing Information Relationships. The purpose of the discussion that follows is to justify our second solution approach for addressing the problem of decentralized particle filtering. As before, the more rigorous presentation of key mathematical concepts will follow the conceptual preview of this section.

A large majority of the current Bayesian state-of-the-art methods for addressing conservative data fusion make use of information divergences in some fashion. A representative illustration would be the Covariance Intersection algorithm [286] presented in Section 2.8.2.4, which ensures that a conservative estimate is obtainable (under the assumptions of the algorithm) by optimizing, with respect to a parameter ω, the trace or the determinant of the resulting fused information matrix. The CI algorithm has enjoyed varying degrees of success when applied to an assortment of data fusion problems [81], [242], [280], [59], [268].

In spite of the numerous documented uses of the CI algorithm, there are known flaws. For example, it tends to be overly conservative in its estimates [103]. The algorithm is restricted by its reliance on the fact that the densities under consideration must be adequately described as being Gaussian in nature. The restrictive constraints of the CI algorithm have been relaxed by the results obtained through the independent investigations of Mahler [181] and Hurley [121]. Their efforts have resulted in a more general form which is more amenable to analysis using differential geometric tools.

To summarize the CI reformulation, Mahler made the keen observation that under the Gaussian assumption, the inflating of the fused information matrices by the CI algorithm was equivalent to allowing the power of the Gaussian function to vary according to $\omega \in [0,1]$ (as opposed to imposing the constraint $\omega = 1$) and normalizing, a process described mathematically by

$$\frac{\mathbf{p}(\mathbf{x})^{\omega}}{\displaystyle\int_{\mathbf{x}\in\mathcal{X}} \mathbf{p}(\mathbf{x})^{\omega}\,\mathrm{d}\mathbf{x}} = \mathcal{N}\left(\mathbf{x};\hat{\mathbf{x}},\frac{\mathbf{P}}{\omega}\right), \qquad \omega \in [0,1]. \tag{3.20}$$

Furthermore, Hurley made the observation that Equation (3.20) could be extended to the case with multiple probability densities, and also that the Gaussian restriction could be removed. If the constraint on the exponent was extended such that the collection of exponents were required to all be positive and sum to 1, then in the two density case one would obtain the following expression

$$\frac{\mathbf{p}_1(\mathbf{x})^{\omega}\mathbf{p}_2(\mathbf{x})^{(1-\omega)}}{\displaystyle\int_{\mathbf{x}\in\mathcal{X}} \mathbf{p}_1(\mathbf{x})^{\omega}\mathbf{p}_2(\mathbf{x})^{(1-\omega)}}, \qquad 0 \leqslant \omega_i \leqslant 1 \quad \text{and} \quad \sum_{i=1}^{N}\omega_i = 1, \tag{3.21}$$

which is exactly the definition of the NWGM given in Equation (3.4). Subsequently, Equation (3.4) also provided a link between the NWGM and the Chernoff information given in Equation (3.5). The benefits of the generalized Chernoff fusion over the Covariance Intersection have been documented in the literature [99]. A considerable amount of effort in recent literature has been spent on trying to obtain meaningful approximations to make solving Equation (2.113) more practical, e.g. see [99].

Establishing the connection between popular Bayesian decentralized data fusion techniques is a necessary task for being able to relate existing algorithms to algorithms formulated on the unit hypersphere. A particularly useful observance is the geometrical interpretation of the Chernoff information, an interpretation that has been pointed out by numerous authors [6, 72, 139, 259]. The information and geometric interpretations associated with state-of-the-art Bayesian decentralized data fusion methods establishes a *relationship baseline* from which we can begin addressing the shortcomings of the gradient-based optimization approach.

3.7.2 Conceptual Motivation. The differential geometry of the unit hypersphere has offered valuable insight into how to formulate fusion strategies on alternative spaces to the usual Euclidean space. Under the locality constraints presented in Section 3.4, both Karcher and Kendall establish conditions under which fusion solutions can be assured to not only exist, but also be unique. Existence and uniqueness assurances are both luxuries that, in general, are difficult if not impossible to establish in general optimization problems.

There is increased analytical value in observing that taking the square-root of a probability density function will produce a sample point along a geodesic on the unit hypersphere. The analytical tools available for defining and manipulating geodesics on the unit hypersphere make it quite simple to generate a collection of samples along the geodesic connecting points ψ_1 and ψ_i. Equivalently, one can calculate a series of geodesics along the same exponential map. Each new geodesic, in the generated series, will be appended to the existing collection of geodesics, a process that is repeated until the entire distance defined by the exponential mapping operator is covered. This process is referred to as geodesic interpolation in Section 3.7.3.

Now, given a collection of sample ψ's, coupled with the constraint that the functions residing on the unit hypersphere must be positive functions, one can show that the representation for each ψ in the original Euclidean space of proper density functions will be a unique representation. The ability to establish conditions under which the properties of existence and uniqueness can be guaranteed allows the following conjectures to be made.

Previously, it has been proven that the existence of geodesics on the surface of the unit hypersphere is guaranteed [225, 226, 299]. The guaranteed geodesic existence, coupled with the existence and uniqueness guarantees for an individual ψ, one is assured that there will exist a ψ along the geodesic that corresponds to a unique probability density function in the originating Euclidean space of proper probability densities. Establishing the uniqueness of available probability density functions is certainly a step in the right direction. The next task is to establish the link between the availability of a unique probability density with the process for generating geodesics on the unit hypersphere. Ultimately, we look to establish the relationship with the process for selecting a particular ψ along the geodesic.

The method for generating geodesics on the unit hypersphere have previously been established in Section 2.3. For example, analytical tools such as exponential and logarithmic mappings were utilized in the gradient-based approach presented earlier. In fact, the parameterized version of the exponential mapping given in Equation (3.17) has direct ties to the Normalized Weighted Geometric Mean presented in Section 3.2, and hence direct ties to the Chernoff information.

3.7.3 Performing Geodesic Interpolation.

The process begins by determining the tangent vector $\boldsymbol{\tau}$ between the local agent's square-root density ψ_1 and a reconstructed density from received data provided by a neighboring agent ψ_i by first defining

$$\begin{aligned} \psi_1 &= \sqrt{\mathbf{p}_1\left(\mathbf{x}_k^f \mid \mathbf{Z}^{k-1}, \mathbf{z}\right)} \\ \psi_i &= \sqrt{\mathbf{p}_i\left(\mathbf{x}_k^f \mid \mathbf{Z}^{k-1}, \mathbf{z}\right)}, \end{aligned} \tag{3.22}$$

then calculating

$$\boldsymbol{\tau}_i = \mathrm{LogMap}_{\psi_1}\left(\psi_i\right). \tag{3.23}$$

Once the tangent vector is defined, by use of the logarithmic mapping operation, the process proceeds by generating the geodesic that connects ψ_1 and ψ_i. The number of square-root densities or how densely one wants to sample the geodesic is a parameter value that needs to be determined prior to implementation. For example, in the simulations carried out in the course of this research it was determined that 30 samples or 30 square-root densities provided more than adequate results for the particular problem studied.

To help visualize this process, Figure 3.7 shows two probability density functions (each could represent a potential landmark state). Within an estimation framework, one could think of the probability density function labeled $\mathbf{p}_1$ in Figure 3.7 as the probability density of a single coordinate state for an environmental landmark that is the result of a Bayesian temporal propagation. Likewise, the probability density function labeled $\mathbf{p}_i$ in Figure 3.7 could potentially be a probability density function that represents a measurement taken of the single landmark state. The task that one would want to perform in this scenario would be to project the densities onto the unit hypersphere, and generate the geodesic that connects the two probability densities. The tool for generating the geodesic would be the parameterized version of the exponential mapping given in (3.17). Every

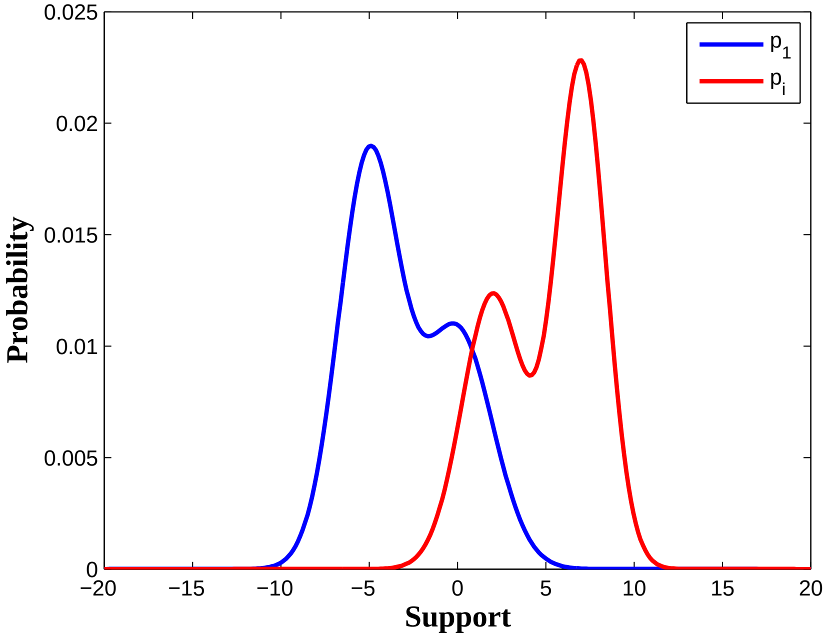


Figure 3.7: Two potential landmark state probability density functions

tenth sample of the geodesic generated using Equation (3.17) for the probability densities in Figure 3.7 is shown in Figure 3.8, where a total of 100 samples were generated. Every tenth probability density is shown for the clarity of the presentation. Note the densities shown in Figure 3.8 represent the densities that have been projected back into Euclidean space already and are not the actual square-root densities.

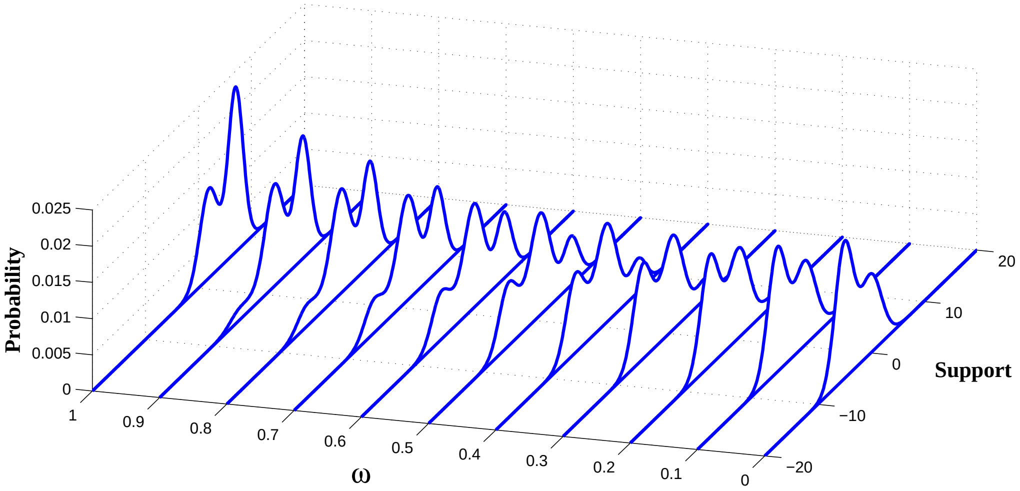


Figure 3.8: One hundred sample densities were generated to form the geodesic between the two initial probability density functions given in Figure 3.7. Every tenth probability density function is shown.

3.7.4 Information Distance Concepts for Fusion. The original square-root densities ψ_1 and ψ_i along with the square-root densities that comprised the geodesic calculated in Section 3.7.3 need to be projected back into the original Euclidean space. The projection is accomplished in the same fashion as in Section 3.6.7, by use of the projection operator given in Equation (3.19). Recall that in Section 2.8.2.6, the Chernoff information was identified as an information measure that was capable of providing density estimates that were conservative when performing decentralized data fusion. In fact, as stated by Hurley [121] and described in [68], the Chernoff information provides the *optimal* achievable exponent in the Bayesian probability of error. Optimal, in this context, refers to the selection of a probability density that is equidistant between two probability densities with respect to the Kullback-Leibler divergence

$$\mathrm{D_{KL}}(\mathbf{p}_{*}\|\mathbf{p}_1) = \mathrm{D_{KL}}(\mathbf{p}_{*}\|\mathbf{p}_2), \tag{3.24}$$

where $\mathbf{p}_*$ is the probability density that satisfies Equation (3.24). The next question needing addressed is just how to select the density $\mathbf{p}_*$, or any other desired density for that matter.

3.7.5 Fusing ψ's for Density Selection. In the absence of an optimal solution in multi-agent decentralized data fusion, one must resort to sub-optimal methods. Information measures have become a popular tool for sub-optimal methods.

The selection of the density that is equidistant in terms of Kullback-Leibler divergences, as is the case in Equation (3.24), provides inspiration for the selection of a fused ψ. The definition given in Equation (3.24) uses the meaning of the term middle as starting from the end points of the geodesic and working towards the *middle*. Another meaning of the term middle can be expressed by arbitrarily selecting the density that resides at the position $\omega = 0.5$. This is equivalent to selecting the density that corresponds to the Bhattacharyya distance. The primary reason for selecting a middle density derives from the desire to select a density based on a divergence measure that has the property of symmetry. In general, the Kullback-Leibler density is not symmetric, but it does have a popular symmetric form given by the Jeffreys Divergence [130], given here as

$$\begin{aligned} \mathrm{D_J}(\mathbf{p}\|\mathbf{q}) &= 0.5\Bigg(\sum_{i=1}^{N}(\mathbf{p}_i - \mathbf{q}_i)\ln\left(\frac{\mathbf{p}_i}{\mathbf{q}_i}\right)\Bigg) & (3.25) \\ &= 0.5\Big(\mathrm{D_{KL}}(\mathbf{p}\|\mathbf{q}) + \mathrm{D_{KL}}(\mathbf{q}\|\mathbf{p})\Big), & (3.26) \end{aligned}$$

which amounts to calculating the arithmetic mean of the component Kullback-Leibler divergences. The Jeffreys divergence does provide some difficulties when considered in the context of density selection. As pointed out by Kailath [146], Basserville [29], and

Johnson *et al.*, [259], the relationship between Jeffreys divergence and the Chernoff information through Steins Lemma [68] is quite a bit more laborious than other divergences. The relationship between Jeffreys divergence and Chernoff information has little significance in itself. However, when viewed within the context of relating existing decentralized data fusion methods with our method, it takes on increased significance.

Given that the Jeffreys divergence is the result of taking the arithmetic mean of the component Kullback-Leibler divergence, it is natural to then ask the question are there any other types of means that could be useful for fusing and selecting particular densities [135]? The most obvious alternate mean values are the geometric and harmonic means, given their fundamental relationship with the arithmetic mean [231], expressed through the so-called Arithmetic-Geometric-Harmonic mean inequality, often abbreviated by AM-GM-HM and is given by

$$HM \leqslant GM \leqslant AM. \tag{3.27}$$

A new divergence measure has recently been presented in the instrumental works of Sinanovic and Johnson [135], [134], and is given the name Resistor divergence as it is obtained by

$$\mathrm{D}_{\mathrm{RE}}(\mathbf{p}\|\mathbf{q})^{-1} = \mathrm{D}_{\mathrm{KL}}(\mathbf{p}\|\mathbf{q})^{-1} + \mathrm{D}_{\mathrm{KL}}(\mathbf{q}\|\mathbf{p})^{-1}. \tag{3.28}$$

As pointed out by Sinanovic and Johnson, Equation (3.28) is equivalent to the harmonic sum of the component Kullback-Leibler divergences, which is equivalent to half of their harmonic mean. The Resistor divergence enjoys several key attributes which make an ideal candidate for density selection. First, it is comprised entirely of component Kullback-Leibler divergences, making it attractive from a computational perspective (provided the component Kullback-Leibler divergences are defined). Second, the Resistor divergence upper-bounds the Chernoff information, meaning that since the Chernoff information represents the best achievable Bayesian probability of error exponent [68], then selecting a density based on the Resistor distance is guaranteed to never produce a probability density that yields an erroneously reduced error exponent. Third, the Resistor divergence provides a symmetric divergence measure by interpreting the concept of *middle* differently than the Chernoff interpretation. In Equation (3.24), the Chernoff information produces a probability density function that is equidistant in terms of component Kullback-Leibler divergences. The Resistor divergence essentially reverses the order of the probability density functions used to calculate the component Kullback-Leibler divergences according to the following definition

$$\mathrm{D}_{\mathrm{KL}}(\mathbf{p}_1 \| \mathbf{p}_*) = \mathrm{D}_{\mathrm{KL}}(\mathbf{p}_2 \| \mathbf{p}_*), \tag{3.29}$$

where $\mathbf{p}_*$ is the probability density that validates Equation (3.29). Stated in words, Sinanovic and Johnson provided the interpretation that instead of determining the middle by starting at the end points of the geodesic and working inwards towards the middle, the Resistor divergence starts in the middle and works outward towards the ends of the geodesic [134].

In an attempt to solidify some of the useful relationship between the Kullback-Leibler divergence, various mean representations, Bhattacharyya divergence, and the Chernoff information, the following example presentation is offered, and is an adaptation of a presentation found in Sinanovic and Johnson [135], [134]. Figure 3.9 shows two probability density functions used to generate Figure 3.10, an adaptation of a figure from [134].

As one might expect, Figure 3.10 clearly demonstrates that the Bhattacharyya distance occurs when the optimization parameter takes on the value $\omega = 0.5$. Also, the Chernoff distance occurs when $\omega = 0.4250$, the maximum value of the geodesic curve. Furthermore, shown in Figure 3.10 is a clear depiction of how the Kullback-Leibler divergence is a non-symmetric function. The derivative of the geodesic curve at the end points represent the Kullback-Leibler divergences given by $\mathrm{D}_{\mathrm{KL}}(\mathbf{p}\|\mathbf{q})$ and $-\mathrm{D}_{\mathrm{KL}}(\mathbf{q}\|\mathbf{p})$ respectively. The intersection of the derivatives occurs at the ω value that corresponds to the density that satisfies Equation (3.29).

A particularly valuable tool offered by Johnson *et al.*, [135], [134] is described with Figure 3.10. In Figure 3.10, the vertical axis represents values obtained from the evaluation of the functional that is optimized in the definition of the Chernoff divergence given

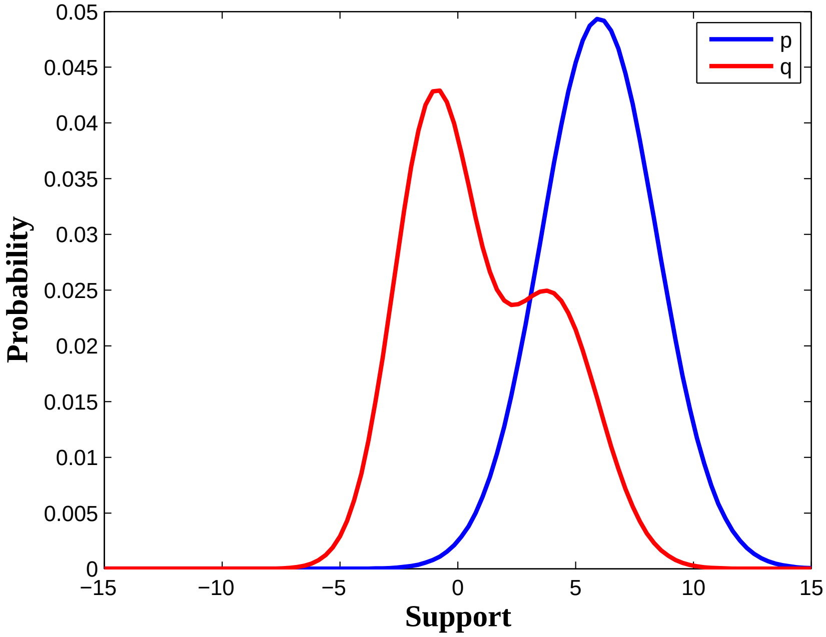


Figure 3.9: Two probability density functions used in the creation of Figure 3.10.

in Equation (2.113) and stated again here for convenience,

$$-\log\left(\int \mathbf{p}(x)^{\omega}\mathbf{q}(x)^{1-\omega}dx\right). \tag{3.30}$$

The curved blue line represents the geodesic that connects two densities $\mathbf{p}_1(\mathbf{x})$ and $\mathbf{p}_2(\mathbf{x})$ and is defined according to

$$\gamma_{\omega}(x)\frac{\mathbf{p}_1(x)^{\omega}\mathbf{p}_2(x)^{1-\omega}}{\int \mathbf{p}_1(x)^{\omega}\mathbf{p}_2(x)^{1-\omega}dx}, \qquad \text{with} \quad 0 \leqslant \omega \leqslant 1. \tag{3.31}$$

The vertical lines that run from the individual values defined by each of the information divergences to the horizontal axis that is defined by the parameter ω represent the value of ω that when plugged into Equation (3.31) results in the respective information divergence value. Summarily stated, the ordering of the various information measures along the vertical axis will remain unchanged regardless of what the individual probability densities under consideration might be. Clearly, the type, shape, and support of various probability density functions define the numerical values given to the information measures in Figure 3.10; however the relationship in terms of where along the vertical axis they are defined will be unchanged. The constant ordering of the information measures in Figure 3.10 assures that the Resistor divergence will always upper bound the Chernoff information regardless of the individual probability densities considered. Additionally, the Chernoff information will always be lower bounded by the Bhattacharyya distance.

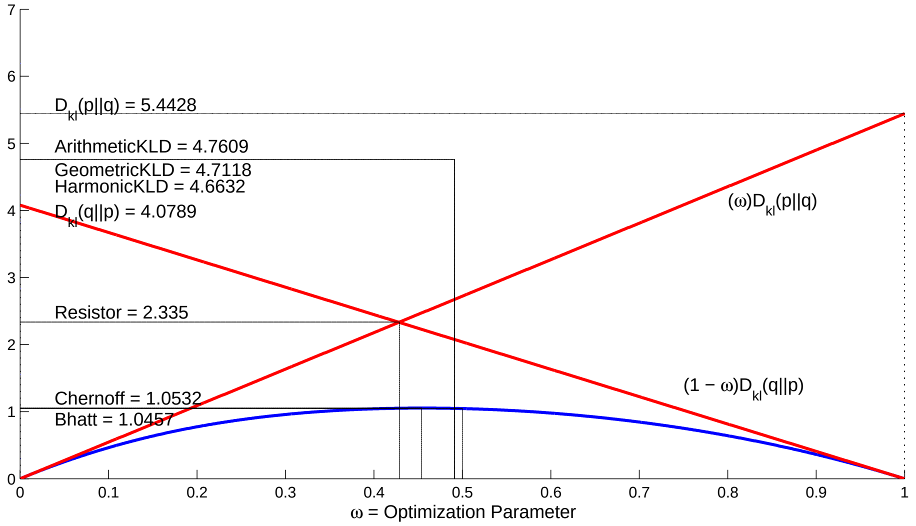


Figure 3.10: Relationships between various information divergences [134]

To summarize this section, the key points of emphasis were the identification of both an upper and lower bound on the Chernoff information. Both bounds are computationally efficient and can be directly related to the use of information measures in existing decentralized data fusion methods. Hence, the goal of density selection is to select the density that adheres to both bounds as the newly fused density to be used in the updating of the local particle collection. The algorithm outlined in this section is shown in the block diagram presented in Figure 3.11.

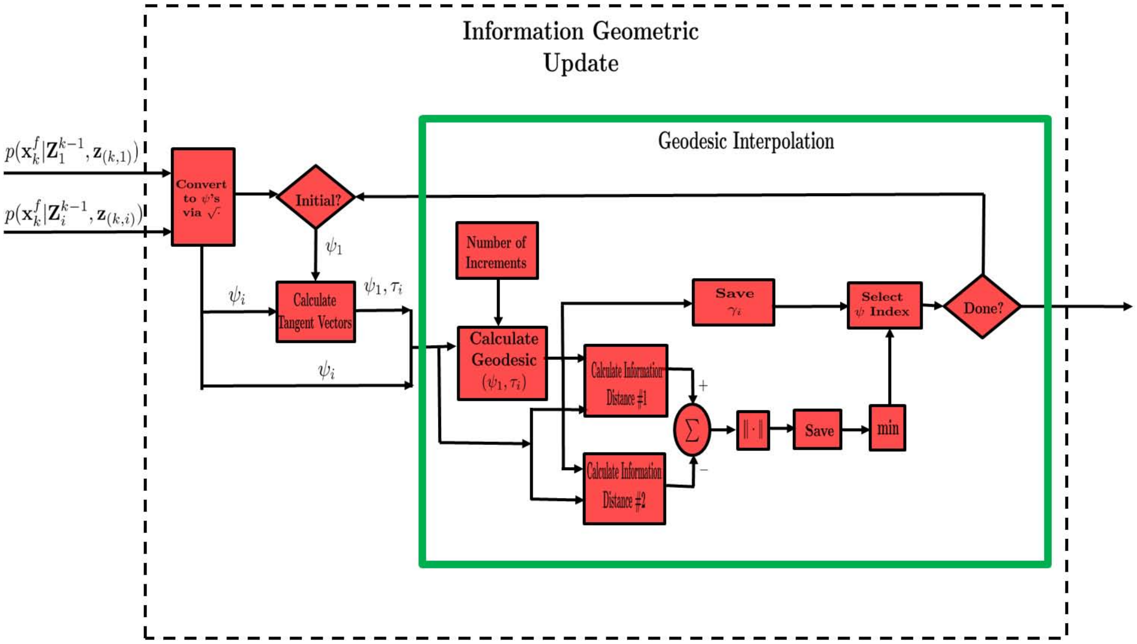

Information Geometric Update
Geodesic Interpolation
$p(\mathbf{x}_k^f|\mathbf{Z}_1^{k-1},\mathbf{z}_{(k,1)})$
$p(\mathbf{x}_k^f|\mathbf{Z}_i^{k-1},\mathbf{z}_{(k,i)})$
Convert to ψ's via $\sqrt{\cdot}$
Initial?
ψ_1
ψ_i
Calculate Tangent Vectors
ψ_1, τ_i
ψ_i
Number of Increments
Calculate Geodesic (ψ_1, τ_i)
Save γ_i
Select ψ Index
Done?
Calculate Information Distance #1
Calculate Information Distance #2
+
$\sum$
−
$\|\cdot\|$
Save
min

Figure 3.11: Second decentralized geometric particle filtering algorithm

3.8 Updating Particle Weights

In this section two separate particle weight update procedures are presented. The first procedure is similar to standard particle filtering weight update approaches discussed in Section 2.5.3 and in [46] with a few differences. The second procedure utilizes the results of the data fusion procedure on the unit hypersphere directly.

3.8.1 Weight Update With MAP Estimation. The likelihood function for the collection of particles is used for the purpose of updating the particle weights. In order to update the weights according to the newly fused density, a few tasks must be accomplished. First, is the calculation of the squared distance of the particle collection. This is accomplished by first calculating the maximum a posteriori or MAP estimate along each dimension separately for the landmark states. The parameter vector $\hat{\mathbf{x}}$, of MAP estimates, is then used in the calculation of the squared distance associated with the particle set leading to the following calculation

$$D^{(i)} = \|\hat{\mathbf{x}} - \mathbf{x}^{(i)}\|^2, \qquad \text{for} \quad i \in \{1, 2, ..., N_p\} \tag{3.32}$$

where N_p represents the total number of particles. The distance calculated in Equation (3.32) is then used to calculate the desired likelihood given by

$$\Lambda\left(\hat{\mathbf{x}} \mid \mathbf{x}^{(i)}\right) = \mathbf{exp}\left\{\frac{-\|\hat{\mathbf{x}} - \mathbf{x}^{(i)}\|^2}{\dfrac{1}{N_p}\displaystyle\sum_{i=1}^{N_p} \|\hat{\mathbf{x}} - \mathbf{x}^{(i)}\|^2}\right\}. \tag{3.33}$$

The result from calculating the likelihood function in Equation (3.33) is used to update the particle weights, as in Equation (3.34)

$$\mathrm{w}^i_{(+)} = \frac{\mathrm{w}^i_{(-)} \times \Lambda\left(\hat{\mathbf{x}} \mid \mathbf{x}^{(i)}\right)}{\dfrac{1}{N_p}\displaystyle\sum_{i=1}^{N_p} \mathrm{w}^i_{(-)} \times \Lambda\left(\hat{\mathbf{x}} \mid \mathbf{x}^{(i)}\right)}. \tag{3.34}$$

Next, the particles are resampled according to the new weights determined in Equation (3.34).

This approach suffers from the following flaw under the gradient-based unit hypersphere fusion approach. After maintaining a representation of a structure capable of recreating any of the desired probability density functions, a wealth of potential information is discarded by adopting a squared distance based approach for constructing likelihood functions that are used to update particle weights. The next, alternate procedure will address this problem by utilizing the results of the geodesic interpolation process for data fusion on the unit hypersphere directly.

3.8.2 Weight Update With Geodesic Interpolation. Once the fused probability density has been selected, the probability weights are updated according to the following procedure. First, recalling that the choice of density representation was the histogram, as such, part of the data communicated between agents was the starting bin location in each data dimension used in the fusion process.

Starting with the first dimension, a prespecified number of bins are generated, each of equal bin width. Then the particle values, that were the result of the local filter update cycle, are weighted according to which bin they reside in. After this process has completed for each of the individual data dimensions, the result will be a collection of D one dimensional likelihood values, where D is the total number of data dimensions.

The likelihood values are not held to the same constraints that valid probability density functions are, mainly that they are not required to sum (integrate in the continuous case) to unity, nor are they required to be finite [38], [201]. Recall that, under the assumption of independence, a joint probability function can also be defined as the product of its marginals. There is an analogous relationship among assumed independent likelihood functions, and it is the joint likelihood function can be expressed as the pointwise product of the individual likelihood functions [201] according to

$$\Lambda(f_{x_1}, f_{y_1}, f_{x_2}, f_{y_2}, f_{x_3}, f_{y_3}, ...) = \Lambda(f_{x_1})\Lambda(f_{y_1})\Lambda(f_{x_2})\Lambda(f_{y_2})\Lambda(f_{x_3})\Lambda(f_{y_3}), \tag{3.35}$$

where f_x and f_y are used to identify the x and y positions of a particular landmark respectively, and the subscript number is used to identify which landmark is being considered. The result of the pointwise multiplication of likelihood functions is then used to update the particle weights that existed just after the local particle propagation cycle. The weight update is calculated according to

$$\mathrm{w}^{i}_{(k)} = \frac{\mathrm{w}^{(i)}_{(k-1)}\,\Lambda(f_{x_1}, f_{y_1}, f_{x_2}, f_{y_2}, f_{x_3}, f_{y_3}, ...)}{\sum_{i=1}^{N_p} \mathrm{w}^{(i)}_{(k-1)}\,\Lambda(f_{x_1}, f_{y_1}, f_{x_2}, f_{y_2}, f_{x_3}, f_{y_3}, ...)}. \tag{3.36}$$

The resulting weights are then used to resample the local particles that existed as a result of the most recent local filter measurement update procedure.

3.8.3 Particle Resampling. In general, the ability to generate samples from an arbitrary probability density function is extremely difficult. The primary point of difficulty resides in the fact that one rarely has access to the true density needing to be sampled. As a result of the complications imposed by arbitrary densities, there have been alternative methods developed. Chief among the alternatives is the process known as importance sampling. The importance sampling process, as discussed in Section 2.5.2, suggests drawing samples from a proposal density in order to compute expectations of another density through the procedure of appropriate weighting. A similar process is utilized in both weight update algorithms before returning the fusion results to the input of the next local agent particle filter algorithm iteration.

3.9 Summary

To summarize this chapter, the following key elements are highlighted. First, the space of probabilities was refined by identifying its associated Riemannian structure. Particular emphasis was given to the computation of relevant intrinsic statistics in the refined space. Then two separate data fusion methods were provided, both of which exploited the Riemannian structure of probabilistic space. The primary surface considered for data fusion of probabilities was the unit hypersphere. The unit hypersphere was shown to be ideally suited, under specific constraints, to the data fusion process.

The first algorithm exploited the tangent spaces of projected probability densities, where tangent vectors were calculated and a mean tangent vector was identified through a simple gradient descent algorithm. The second algorithm looked to the inherent relationships between the probabilistic space on the unit hypersphere and the theory of information measures to perform the data fusion process. The second algorithm was shown to be related to existing data fusion methods.

Key distinctions between the proposed algorithms and existing algorithms were stated. Existing procedures can not make use of the simplistic probability density representation offered by histograms because they must perform a division operation. We remove the need to perform the division of the fused density by the common information through the use of the differential geometric relationships defined on the unit hypersphere.

In the next chapter the algorithms are examined thoroughly, and compared to existing data fusion algorithms. Their performances are compared, and differences are highlighted. The analysis is performed through the use of a two dimensional navigation scenario.

IV. Detailed Simulation Analysis and Discussion of Results

4.1 Chapter Overview

The previous chapters have motivated this line of research, presented necessary background material, and developed novel decentralized geometric particle filtering algorithms. In this chapter, an example scenario is presented along with detailed analysis. The decentralized particle filtering algorithms presented in Chapter III are compared with current state-of-the-art fusion techniques, and results are rigorously analyzed. All of the results presented in this chapter were obtained with a detailed simulation developed in the MATLAB® software environment. The majority of the analysis presented in this chapter is with respect to the algorithm presented in Section 3.7. When alternative algorithms are used it will be made explicit. The general models that were used in the simulation environment are presented next, followed by the analysis and results obtained under various operating conditions.

4.2 Simulation Scenarios, Models, and Parameters

4.2.1 Simulation Scenario. Consider the following scenario. Two mobile agents move in an environment comprised of point features that are used as landmarks. Each agent moves in a circular trajectory. A typical scenario can be seen in Figure 4.1. In this particular scenario realization, the 3 point features used are marked using asterisks. The true trajectory of each agent is conveyed through the dashed blue line with the estimated trajectory being the solid red line. The dots located in the particle clouds represent the sample mean of each respective agent's particles. In the two-dimensional scenario shown in Figure 4.1, each agent's motion is characterized by three kinematic states. These states are a cartesian coordinate position (x, y) and a heading angle φ. The agent states can be

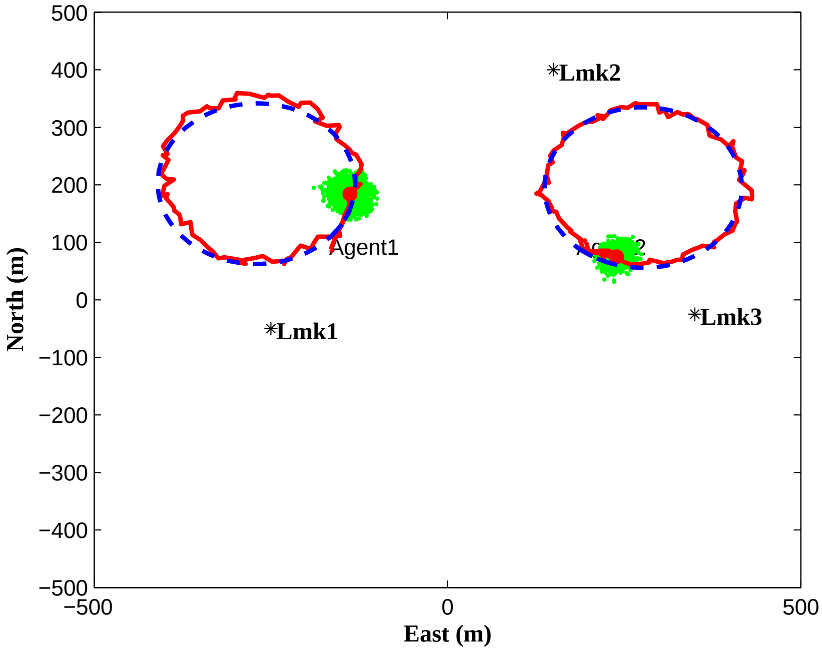


Figure 4.1: Representative scenario considered in the example.

expressed together with the following state vector definition

$$\mathbf{x}_{a,i} = \begin{bmatrix} x_{a,i} \\ y_{a,i} \\ \varphi_{a,i} \end{bmatrix}, \tag{4.1}$$

where the subscript a denotes agent and subscript i denotes which agent. It is common place to see Equation (4.1) referred to as an agent's pose in the robotics literature [243]. As the agents propagate through the environment they can take measurements of landmarks in the form of range and bearings measurements. Each landmark is characterized by a cartesian position expressed as

$$\mathbf{x}_{l,n} = \begin{bmatrix} x_{l,n} \\ y_{l,n} \end{bmatrix}, \tag{4.2}$$

where the subscript l denotes landmark and subscript n denotes which landmark. Given Equations (4.1) and (4.2) the complete state vector considered by each agent is expressed as

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{a,i} \\ \mathbf{x}_{l,n} \end{bmatrix}. \tag{4.3}$$

4.2.2 Process Model. The process model is used to describe the time evolution of the states. In this particular process model, each agent moves with both a constant velocity of $9.5\frac{m}{s}$ and a constant turn angle of $4\frac{deg}{sec}$. Additionally, each agent is constrained to propagate in a planar world, meaning that altitude information is deterministic and fixed. Each feature is assumed to be stationary but a slight amount of process noise is added at each filter iteration for stability. There are several known process models that would be adequate in describing this scenario, which is expressed in general form by

$$\begin{bmatrix} x_{a,i} \\ x_{l,n} \end{bmatrix} = \begin{bmatrix} f(\mathbf{x}_{a,i}, u) \\ \mathbf{x}_{l,n} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\omega_{a,i}} \\ \boldsymbol{\omega_{l,n}} \end{bmatrix}, \tag{4.4}$$

where u denotes a control input and $\boldsymbol{\omega_{a,i}}$ and $\boldsymbol{\omega_{l,n}}$ represent agent and landmark process uncertainty respectively. The process noise terms are assumed to be zero mean white Gaussian noise, i.e.

$$\mathrm{E}[\boldsymbol{\omega}(k)] = \mathbf{0}, \qquad \forall k \tag{4.5}$$

with covariance

$$\mathrm{E}[\boldsymbol{\omega}(k)\boldsymbol{\omega}^T(k+t)] = \begin{cases} \mathbf{Q}(k), & \text{if} \quad t = 0; \\ 0, & \text{otherwise.} \end{cases} \tag{4.6}$$

The continuous time kinematics model used to propagate the agents in the simulation is known in the literature as the unicycle model [227] and given by

$$f(\mathbf{x}_{a,i}, u) = \begin{cases} \dot{x}(t) = V(t)\cos(\varphi(t)), \\ \dot{y}(t) = V(t)\sin(\varphi(t)), \\ \dot{\varphi}(t) = S(t), \end{cases} \tag{4.7}$$

where $V(t)$ and $S(t)$ represent the translational and rotational velocities respectively, and comprise the control inputs. Given Equation (4.7) and the stationary feature assumption, the unified discrete time process model employed in the simulation took the following form

$$\begin{bmatrix} x_{a,i}(k) \\ y_{a,i}(k) \\ \varphi_{a,i}(k) \\ x_{l,1}(k) \\ y_{l,1}(k) \\ x_{l,2}(k) \\ y_{l,2}(k) \\ x_{l,3}(k) \\ y_{l,3}(k) \end{bmatrix} = \begin{bmatrix} x_{a,i}(k-1) + V(k)\cos(\varphi_{a,i}(k-1))\Delta t \\ y_{a,i}(k-1) + V(k)\sin(\varphi_{a,i}(k-1))\Delta t \\ \varphi_{a,i}(k-1) + S(k)\Delta t \\ x_{l,1}(k-1) \\ y_{l,1}(k-1) \\ x_{l,2}(k-1) \\ y_{l,2}(k-1) \\ x_{l,3}(k-1) \\ y_{l,3}(k-1) \end{bmatrix} + \begin{bmatrix} \boldsymbol{\omega}_{\boldsymbol{x_{a,i}}}(k) \\ \boldsymbol{\omega}_{\boldsymbol{y_{a,i}}}(k) \\ \boldsymbol{\omega}_{\boldsymbol{\varphi_{a,i}}}(k) \\ \boldsymbol{\omega}_{\boldsymbol{x_{l,1}}}(k) \\ \boldsymbol{\omega}_{\boldsymbol{y_{l,1}}}(k) \\ \boldsymbol{\omega}_{\boldsymbol{x_{l,2}}}(k) \\ \boldsymbol{\omega}_{\boldsymbol{y_{l,2}}}(k) \\ \boldsymbol{\omega}_{\boldsymbol{x_{l,3}}}(k) \\ \boldsymbol{\omega}_{\boldsymbol{y_{l,3}}}(k) \end{bmatrix}, \tag{4.8}$$

where the subscript i identifies which agent, and the subscripts a and l denote agent and landmark states respectively.

4.2.3 Measurement Model.

Measurements of the state of the system are made according to a nonlinear measurement equation of the generic form

$$\mathbf{z}(k) = \mathbf{h}(\mathbf{x}(k)) + \mathbf{v}(k), \tag{4.9}$$

where $\mathbf{z}(k)$ is the actual measurement made at time k, $\mathbf{x}(k)$ is the state at time k, $\mathbf{v}(k)$ is additive white Gaussian measurement noise, and $\mathbf{h}(\cdot, k)$ is a nonlinear measurement model that maps the current state to the measurement space [193]. In the simulation environment, each agent had access to either range measurements, bearing measurements, or both from local sensors. The range and bearing measurement model used is defined in Equation

(4.10) as

$$\mathbf{z}(k) = \begin{bmatrix} \sqrt{(x_{l,n}(k) - x_{a,i}(k))^2 + (y_{l,n}(k) - y_{a,i}(k))^2} \\ \arctan\left(\frac{y_{l,n}(k) - y_{a,i}(k)}{x_{l,n}(k) - x_{a,i}(k)}\right) - \varphi_{a,i}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{v}_r(k) \\ \mathbf{v}_\theta(k) \end{bmatrix}, \tag{4.10}$$

where the $\arctan$ function in Equation 4.10 is the 4 quadrant version. Similar to the process noise, the measurement noise is considered to be zero mean white Gaussian noise, hence

$$\mathrm{E}[\mathbf{v}(k)\mathbf{v}^T(k+m)] = \begin{cases} \mathbf{R}(k), & \text{if} \quad m = 0; \\ 0, & \text{otherwise.} \end{cases} \tag{4.11}$$

4.3 Setting the Stage: Parameters & Assumptions

Unless otherwise stated, the results presented used the parameter values in Table 4.3, along with the assumptions and conditions that follow.

The local propagation of the agent kinematics in the particle filter was performed via a first order Euler integration of the differential equations given in Equation (4.7), which is expressed in Equation (4.8). The measurements made available to the agents by their local sensors were both range and bearing measurements according to Equation (4.10). The parameters listed in Table 4.3 were used by all of the agents. Furthermore, the process noise covariance was assumed to be time invariant such that it was held constant throughout the entire length of a simulation run. Likewise, the measurement noise variances were also assumed to be time invariant and held constant. It was assumed that all of the agents used within any particular simulation run were able to access the same global reference frame.

Communication links between agents were modeled as being stochastically available. The stochastic communication characteristic was produced by first using a random number generator to produce a value between 0 and 1. An availability selection criteria was set to 0.5, then a simple test to determine if the random number generated was greater than or less than the selection criteria was performed. If the number was greater than, then the agents were allowed to communicate. Similarly, if the number was less than the

Table 4.1: Parameters values used in the simulation

Name	Symbol	Value	Units
Simulation Length	T_f	100.0	seconds
Sampling Period	dt	1.0	seconds
Number of Particles	N_p	5000	unitless
Number of Point Features	N_f	3	unitless
Number of Agents	N_a	2	unitless
Initial Agent Position Covariance	$\mathbf{P}_{ag_x,ag_y}$	$\sigma^2 = 15^2$	meters
Initial Agent Heading Covariance	$\mathbf{P}_{ag_\theta}$	$\sigma^2 = 1^2$	degrees
Initial Feature Covariance	$\mathbf{P}_{f_x,f_y}$	$\sigma^2 = 50^2$	meters
Agent Position (X & Y) Process Noise Covariance	$\boldsymbol{\omega_{x,y}}$	$\sigma^2 = 3.0^2$	meters
Agent Heading (φ) Process Noise Covariance	$\boldsymbol{\omega_{\varphi}}$	$\sigma^2 = 7.0^2$	degrees
Feature Position (X and Y) Process Noise Covariance	$\boldsymbol{\omega_{l_{x,y}}}$	$\sigma^2 = 3.0^2$	meters
Range Measurement Covariance	$\mathbf{v}_R$	$\sigma^2 = 15.0^2$	meters
Bearing Measurement Covariance	$\mathbf{v}_\theta$	$\sigma^2 = 10.0^2$	degrees
Number of Geodesic Samples	γ_s	30.0	unitless
Number of Histogram Bins Used	N_b	50.0	unitless

selection criteria, then communication between the agents was prohibited. Figure 4.2 represents the communication availability for a single simulation run where a red bar is used to indicate that an agent was allowed to communicate with the other agent at the associated time. Likewise, the absence of a red bar denotes that communication between agents was not permitted. The use of stochastic communication links was chosen because they were determined to better represent true network communications over the continuous and reliable communication links commonly assumed.

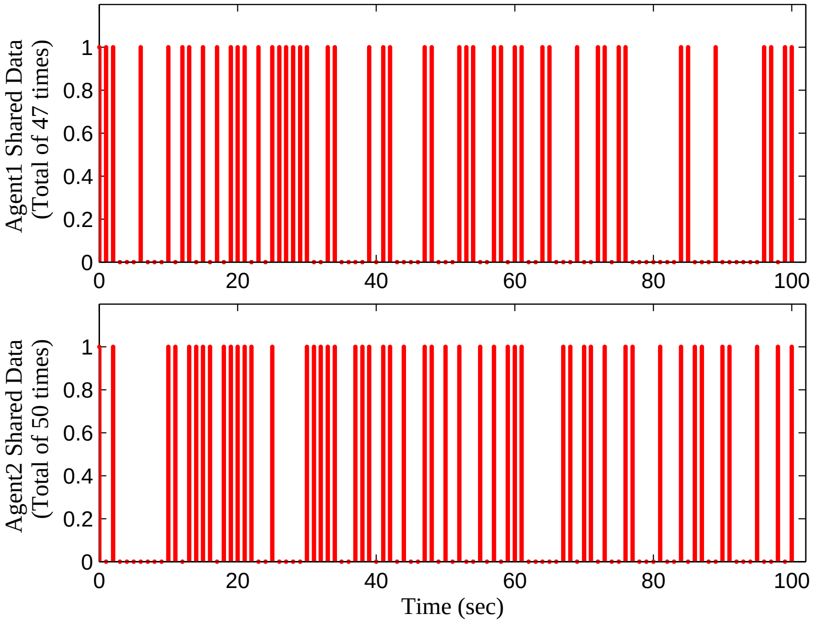


Figure 4.2: Random time epochs when agents communicated

Finally, all of the results shown were obtained via the use of Monte Carlo trials, where the number of trials performed was chosen to be 100. For purposes of clarity, the results for agent1 are primarily presented, and if discussion is necessary regarding the results of other agents as they pertain to a particular situation, it will be done. Otherwise, it is implied that the results produced by other agents are statistically similar to those for the agent presented. Also, the estimation error is calculated according to

$$\text{Error} = \text{Estimate} - \text{Truth}. \tag{4.12}$$

The following can loosely be interpreted as a reading guide for the results and analysis that follows. The topics presented can be thought of as belonging to one of the following four categories. The first category is concerned with the comparison of the two algorithms presented in the previous chapter. The second category is focused on what is called here the operational integrity of the algorithm *i.e.*, does perform as expected under various testing scenarios. Testing scenarios included the algorithm's ability to produce consistent estimates as defined in Equation (4.13), its behavior when the number of agents is changed, its behavior when the number of particles is changed, and other traditional validation procedures commonly found in the decentralized data fusion literature. The next category is focused on the ability of the algorithm to perform in various measurement scenarios. For example, the situation where the agents have access to range and bearing measurements, range only measurements, or bearing only measurements were considered. The final category is a comparison of the proposed algorithm with two current state-of-the-art methods used for decentralized data fusion.

4.4 Comparing Gradient and Information Based Algorithms

In Chapter III two different decentralized particle filtering algorithms were presented. One was based on gradient-based optimization methods and the use of the tangent space of the unit hypersphere, while the other looked to take advantage of the information interpretation that can be given to the unit hypersphere. In this section the two algorithms are compared.

4.4.1 Direct Comparison. For a single run of the unit hypersphere based algorithm, the following results in Figures 4.3 and 4.4 were obtained for agent1's vehicle and landmark states respectively. Both plots show the estimation error in red, and the corresponding filter generated $\pm 1\sigma$ bounds in black. Figures 4.3 and 4.4 represent a

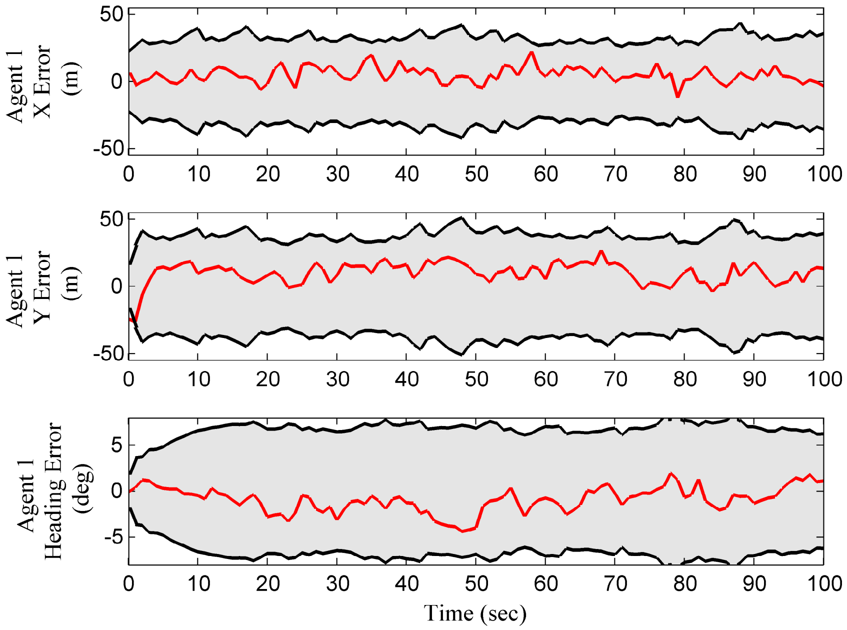


Figure 4.3: Estimation error (red) and the corresponding filter generated $\pm 1\sigma$ bounds (black) for agent1's vehicle specific states

single Monte Carlo trial of the information based algorithm on the unit hypersphere. It can clearly be seen that the fusion algorithm was able to produce estimation errors that are well within the $\pm 1\sigma$ bounds throughout the entire trial for all states. This is an indication that the filter generated standard deviations are slightly pessimistic. Figure 4.5 shows all 100 agent1 x-position estimation errors (blue) obtained in the Monte Carlo trials, along with the mean filter generated $\pm 1\sigma$ bounds (red), the standard deviation of the ensemble estimation errors (yellow), and the mean estimation error (green).

Recall that the reasons for considering the information based approach was to take advantage of the selected densities and to be able to implement a mechanism to help guard against overly optimistic estimation results. Upon initial analysis, the two algorithms surprisingly produced nearly identical results as can been seen in Figures 4.6, 4.7, and 4.8 where the 1σ ensemble error standard deviations are compared for agent1's x-position, y-position, and heading angle respectively. Also included in Figures 4.6, 4.7, and 4.8, for a

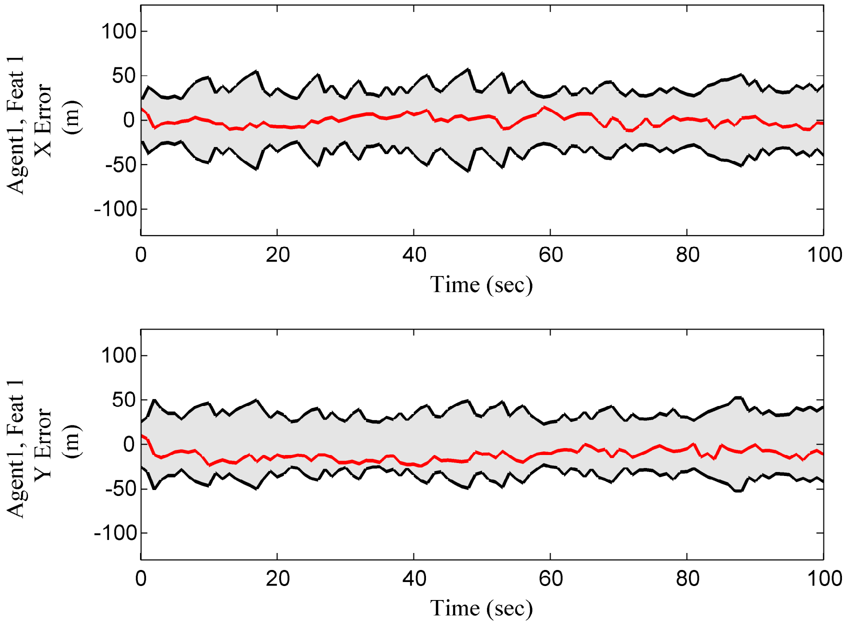


Figure 4.4: Estimation error (red) and the corresponding filter generated ±1σ bounds (black) for agent1's landmark specific states

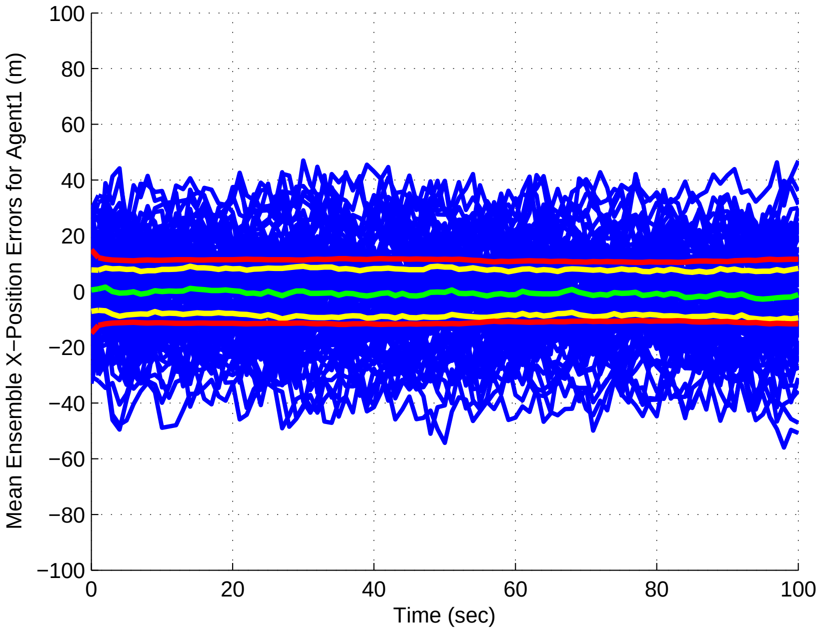


Figure 4.5: Monte Carlo trials (100) showing agent1's x-position state estimation errors (blue), mean filter generated ±1σ bounds (red), the ensemble error ±1σ bounds (yellow), and the mean estimation error (green)

point of reference is the results produced by attempting to produce a *”middle"* probability density without projecting onto the unit hypersphere. As can clearly be seen, the naïve approach is inferior to the proposed algorithms.

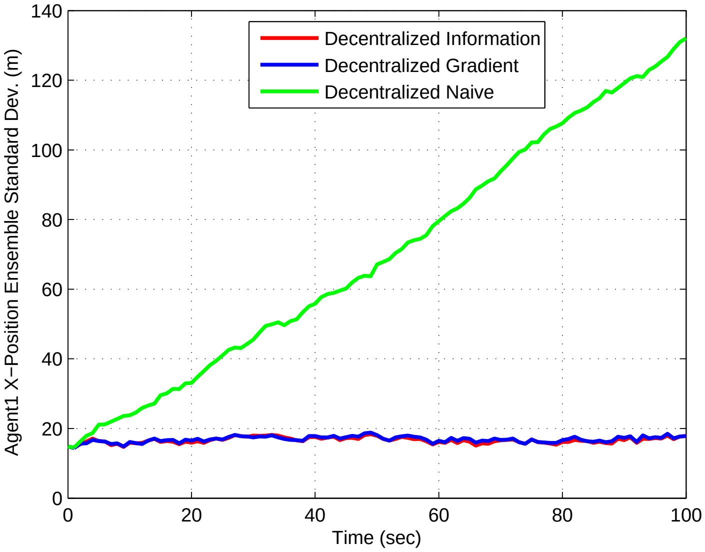


Figure 4.6: Comparison of Agent1’s x-position uncertainty obtained by the gradient-based, information based, and naïve based algorithms in meters (Over 100 Monte Carlo Trials)

4.4.2 A Closer Look. The results produced by both the algorithms given in the previous chapter were surprising because of the reasons for considering the alternative formulation described in Section 3.7. However, upon closer inspection the similarity in the results can be attributed to the facts that the trajectories of both agents were relatively benign, that all of the input noises were time invariant Gaussian noises, and that only map states are communicated between the agents.

The result of the mild dynamics, well modeled Gaussian disturbances, and the sharing of landmark states only was probability densities that were reasonably described as Gaussian being selected in both the gradient-based and information-based global fusion processes. Typical results for the x-position of the first landmark can be seen in Figures 4.9 and 4.10 which are representative single dimensional probability density functions for

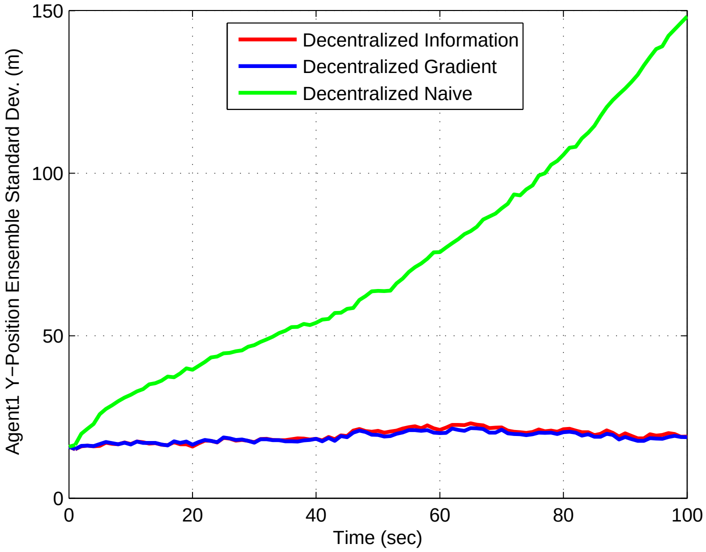


Figure 4.7: Comparison of Agent1's y-position uncertainty obtained by the gradient-based, information based, and naïve based algorithms in meters (Over 100 Monte Carlo Trials)

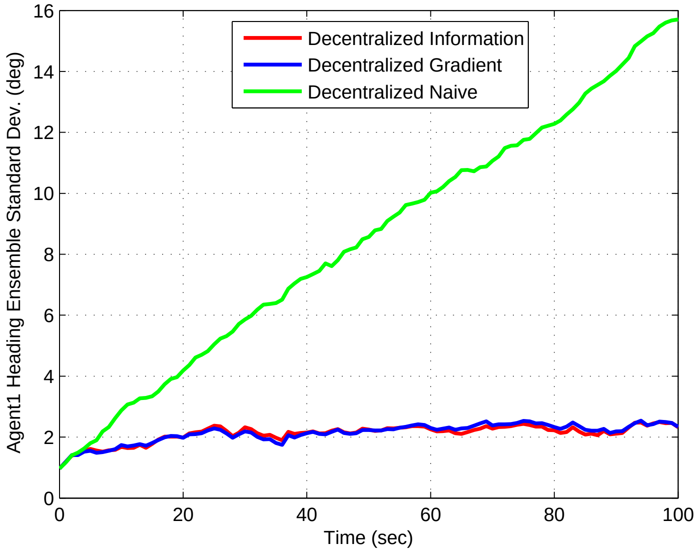


Figure 4.8: Comparison of Agent1's heading uncertainty obtained by the gradient-based, information based, and naïve algorithms in degrees (Over 100 Monte Carlo Trials)

landmark1's x-position obtained by the gradient based and information based algorithms respectively.

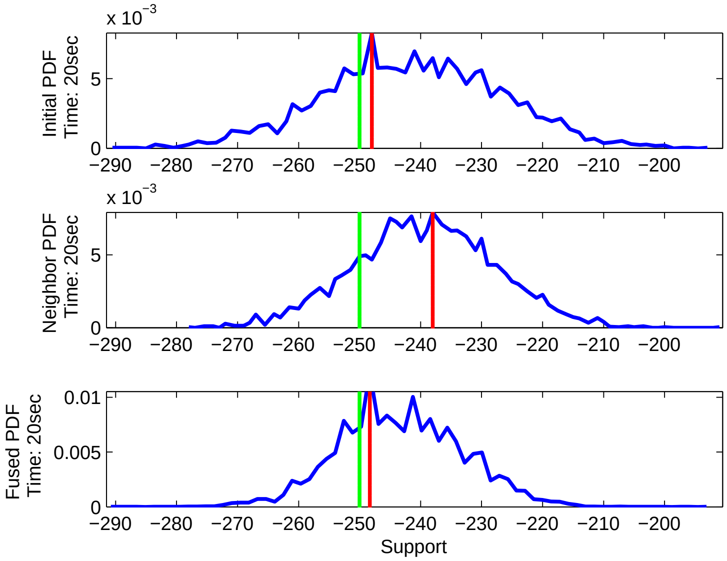


Figure 4.9: Landmark1's x-position pdf from the gradient-based algorithm where the green bar represents the true position and the red bar represents estimated position (100 MC Runs)

4.5 Algorithm Operational Integrity

4.5.1 Ability to Perform Consistent Estimation. The first line of analysis was to determine if the proposed algorithm was capable of producing consistent estimates. For purposes of clarity, the definition of consistent estimates was given in Section 2.8.2.3 as

$$\mathbf{P}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}} - \mathbf{E}[\mathbf{P}_t] \succeq \mathbf{0}, \tag{4.13}$$

where the symbol $\succeq$ was used to express the fact that the left hand side of Equation (4.13) represents a positive semi-definite matrix.

In order to determine if the estimates produced were in fact consistent the following method was used. Given that in the multi-agent scenario with an *ad hoc* network an optimal solution is not available, then the closest to an optimal solution that one can ob-

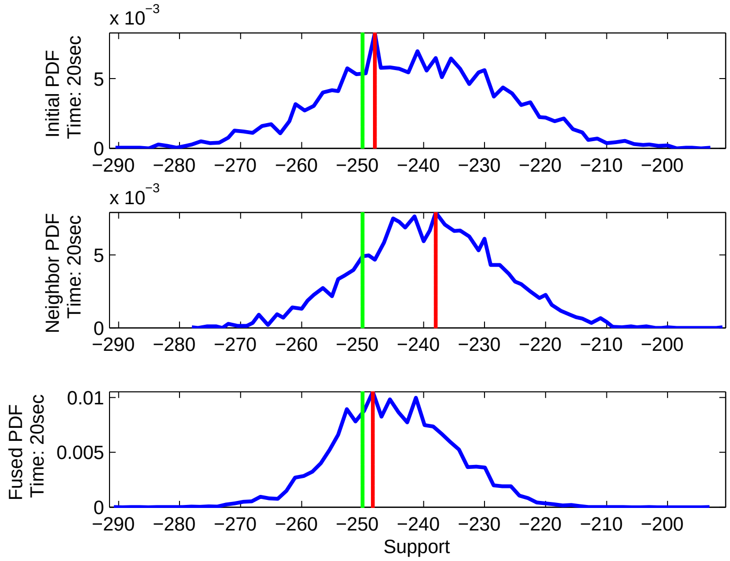


Figure 4.10: Landmark1's x-position pdf from the information-based algorithm where the green bar represents the true position and the red bar represents estimated position (100 MC Runs)

tain is in the centralized fusion case. This means that the uncertainty obtained through the implementation of a decentralized architecture should result in a higher uncertainty than if a centralized architecture was used. So a simple test of consistency would be the ratio between the uncertainties of a particular state resulting from the use of a centralized processing scheme and the use of a decentralized processing scheme *i.e.*,

$$\phi = \frac{\sigma_s^{(c)}}{\sigma_s^{(d)}}, \tag{4.14}$$

where the subscript s is used to declare the particular state under consideration, and the superscripts (c) and (d) identify whether the uncertainty was obtained from the use of a centralized or decentralized processing scheme respectively. If the results of Equation (4.14) are less than one, then the estimate is declared to be consistent. However, if the result produced by Equation (4.14) are greater than one, then the estimate is declared inconsistent. A similar test can be found in the recent works of Nemra *et al.*, [3], the one used here is defined as

$$\phi = \begin{cases} \phi < 1, & \text{Implies that an estimate is consistent;} \\ \phi > 1, & \text{Implies that an estimate is inconsistent.} \end{cases} \tag{4.15}$$

The results for the agent specific states for agent1 over 100 Monte Carlo runs can be seen in Figures 4.11, 4.12, and 4.13 where Figure 4.11 represents agent1's x-position coordinate uncertainty in meters, Figure 4.12 represents agent1's y-position coordinate uncertainty in meters, and Figure 4.13 represents agent1's heading angle uncertainty in degrees.

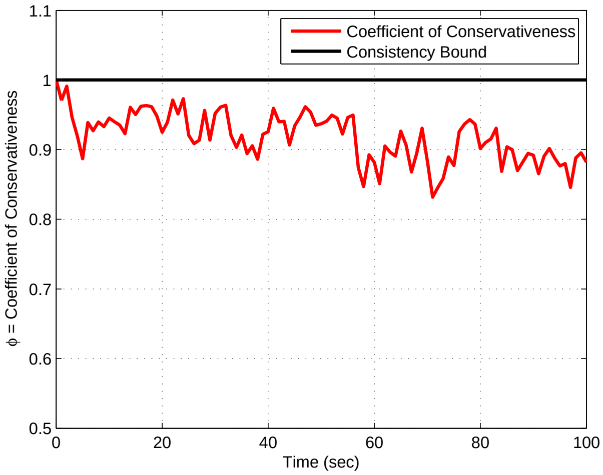


Figure 4.11: Agent1 x-position consistency test (Over 100 Monte Carlo Trials)

Clearly, the decentralized geometric particle filtering algorithm is capable of producing consistent estimates for this problem.

4.5.2 Individual vs. Centralized vs. Decentralized. There are numerous potential benefits offered by a multiple agent network, where the data agents undertake tasks with knowledge of the networks mission, or at least knowledge of a portion of the network, over just a collection of several data agents operating without regard to any other agent. It is likely that tasks can be performed in a more timely manner, and superior estimates can be produce (in a minimum mean square error sense) in the multi-agent decentralized

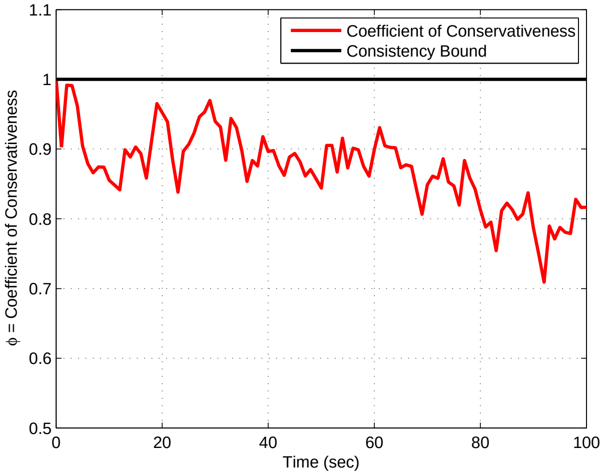


Figure 4.12: Agent1 y-position consistency test (Over 100 Monte Carlo Trials)

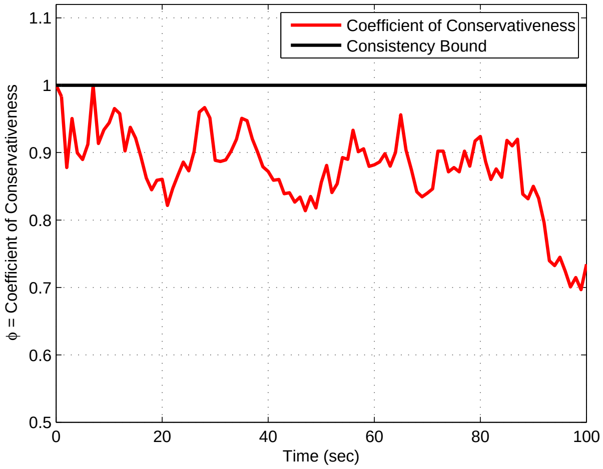


Figure 4.13: Agent1 heading angle consistency test (Over 100 Monte Carlo Trials)

network architecture over the collection of individual agents. The implication here is that agents operating independently should result in a larger estimation error uncertainty than the estimation error uncertainty obtained by agents that operate together in some fashion. Likewise, if a centralized architecture is used, the estimation uncertainty should be smaller than if a decentralized architecture is used to govern the network.

The so-called *estimation error hierarchy* can be seen in Figures 4.14, 4.15, and 4.16. As was the case in Section 4.5.1, the figures show the result of 100 Monte Carlo runs with Figure 4.14 representing agent1's x-position coordinate uncertainty in meters, Figure 4.15 representing agent1's y-position coordinate uncertainty in meters, and Figure 4.16 representing agent1's heading angle uncertainty in degrees. The suggestion that an uncertainty hierarchy exists between processing architectures is validated with the results shown in Figures 4.14, 4.15, and 4.16.

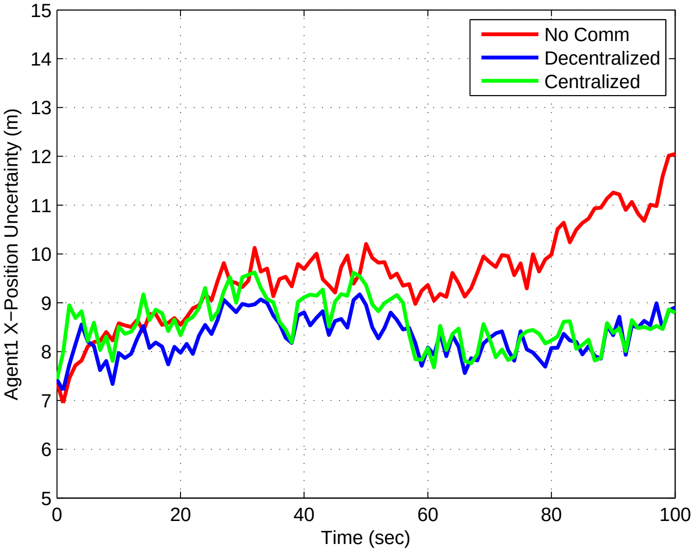


Figure 4.14: Agent1 x-position uncertainty for centralized processing, decentralized processing, and individual processing (Over 100 Monte Carlo Trials)

4.5.3 Impact of Number of Particles. A common problem still needing a more thorough treatment is the number of particles necessary to perform accurate estimation. Certainly the exact number of particles needed will be a function of the application, but

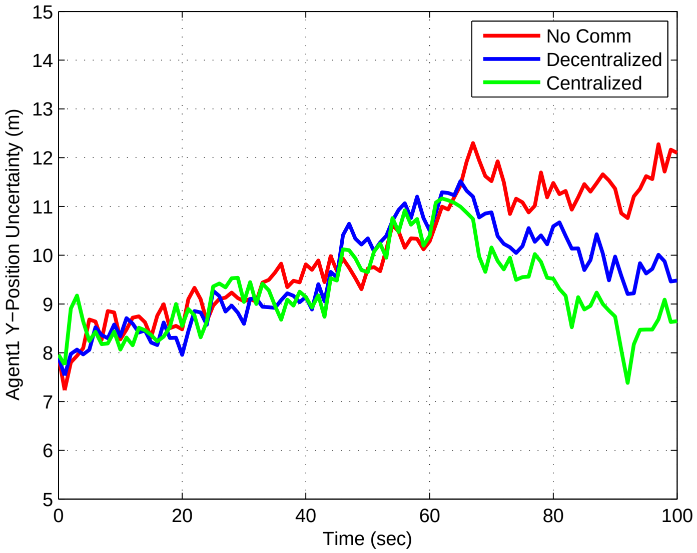


Figure 4.15: Agent1 y-position uncertainty for centralized processing, decentralized processing, and individual processing (Over 100 Monte Carlo Trials)

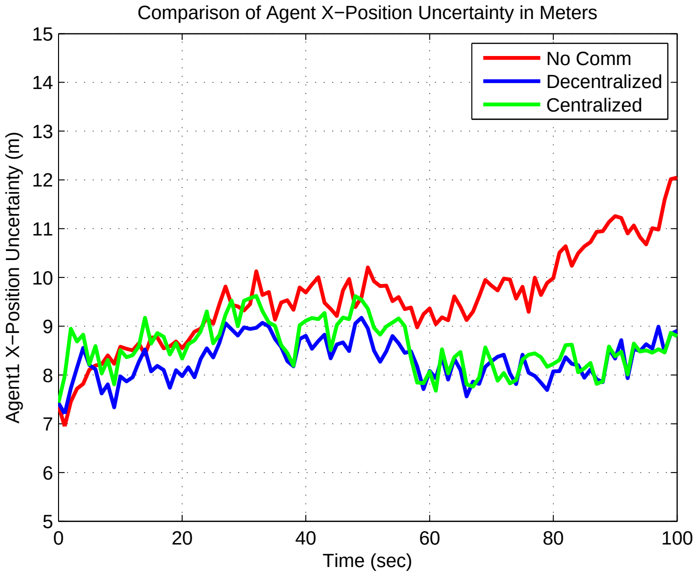


Figure 4.16: Agent1 heading angle uncertainty for centralized processing, decentralized processing, and individual processing (Over 100 Monte Carlo Trials)

a “rule of thumb" is not available. In short, an analogous rule to the Nyquist rate is not available, therefore the number of particles used is typically refined through repeated trial and error, then held constant throughout the length of the filter’s use. It should be noted that there have been some adaptive sample size algorithms presented in the literature. For example, an influential adaptive sample size techniques was introduced by Fox [102] where an adaptive technique based on the use of the Kullback-Leibler divergence to bound the error in the estimate of the true posterior probability density was proposed. Alvaro Soto et al. [262] recognized some shortcomings of the Kullback-Leibler divergence sampling technique, mainly that the samples originate from a proposal density and not from the desired posterior density and proposed improvements.

The fact of the matter is that the number of particles used will have a direct impact on a particle filter’s ability to produce accurate estimates. Certainly, there are applications where the number of particles needed to meet a minimum level of accuracy will change. For example, in an aerial vehicle application when the trajectory can be described as being benign, the number of particles needed will surely be less than in the situation where the trajectory is highly dynamic.

The purpose of this section is not to offer an algorithm for adaptive particle selection. The intent is to acknowledge the impact that the number of particles has on the ability to produce faithful estimates, and that the proposed algorithm does not violate this intuition. As can be seen in Figures 4.17, 4.18, and 4.19, as the number of particles increases, the corresponding Root Mean Square Error (RMSE) in the state estimate decreases. Fundamentally, this is because access to more samples allows the particle filtering algorithm to achieve a more comprehensive representation of the state-space, in addition to a finer resolution of the state-space. From a more particle point of view, one should notice the overall scale and that as the number of particles increases the overall impact on the state uncertainty is not drastic.

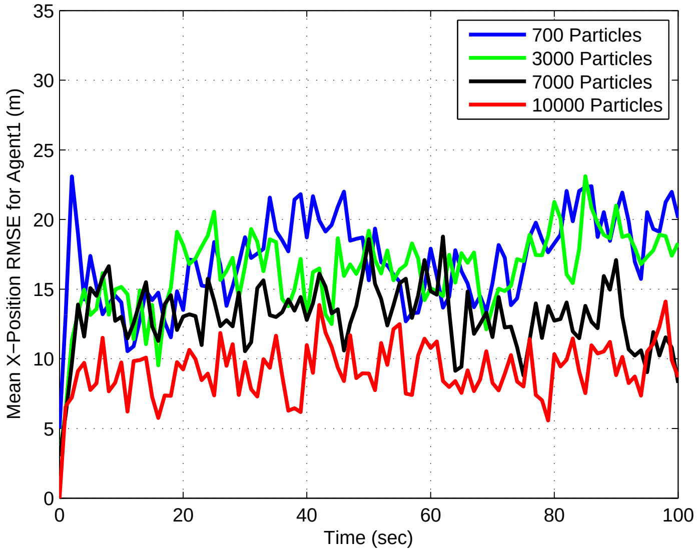


Figure 4.17: Agent1 x-position RMSE estimation accuracy vs. number of particles (Over 100 Monte Carlo Trials)

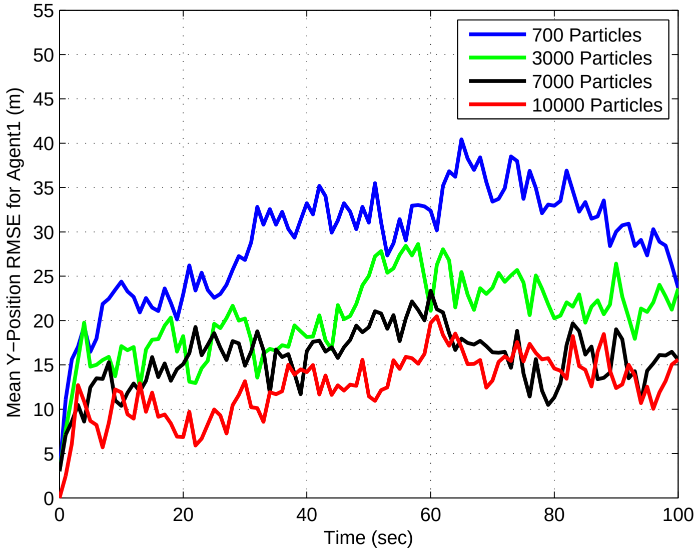


Figure 4.18: Agent1 y-position RMSE estimation accuracy vs. number of particles (Over 100 Monte Carlo Trials)

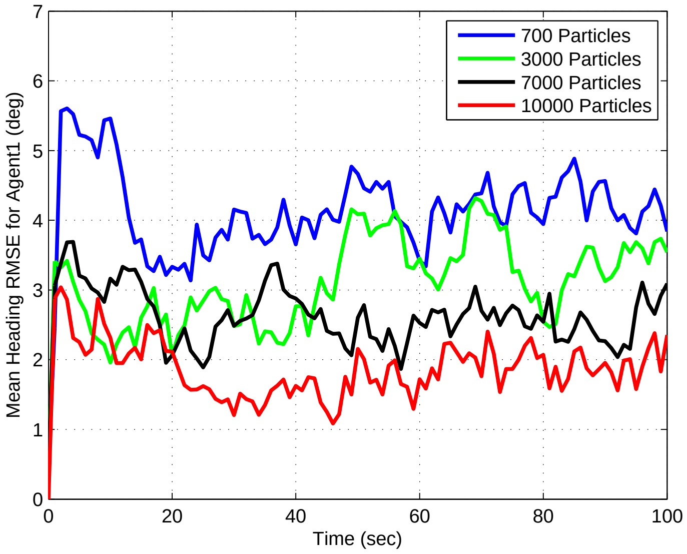


Figure 4.19: Agent1 heading RMSE estimation accuracy vs. number of particles (Over 100 Monte Carlo Trials)

4.5.4 Impact of Number of Agents. In a similar fashion to the analysis performed in Section 4.5.3, the impact of the number of agents used on the accuracy of estimates obtained is explored. Intuitively, as the number of agents increases, one should expect that the uncertainty in the state estimates should decrease. The reason for the uncertainty reduction is due to the added number of measurements made available by the increasing number of agents. Figures 4.20, 4.21, and 4.22 show the estimation uncertainty for agent1's x-position in meters, y-position in meters, and the heading in degrees. As expected, the scenario run with 8 agents produced smaller estimation uncertainties than the scenario with only two agents respectively, albeit modest improvement. The improvement of estimation accuracy as a function of the number of agents can also be seen in the bar plots of Figures 4.23, 4.24, and 4.25 where the height of the bars corresponds to the final uncertainty of the corresponding state. One can clearly see the gradual improvement of the final state uncertainty as the number of agents increases from 2 to 8.

4.5.5 Information Analysis. If the centralized processing architecture is assumed to produce estimation results that are the closest to the true value, then the amount

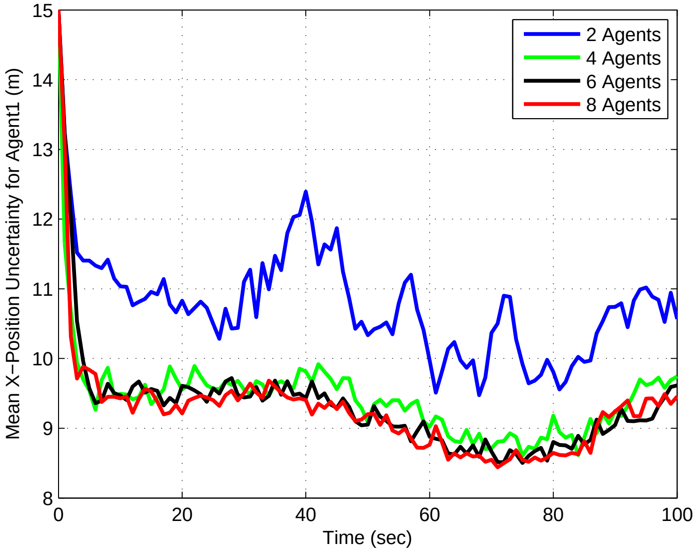


Figure 4.20: Agent1 x-position uncertainty analysis of estimation accuracy vs. number of agents (Over 100 Monte Carlo Trials)

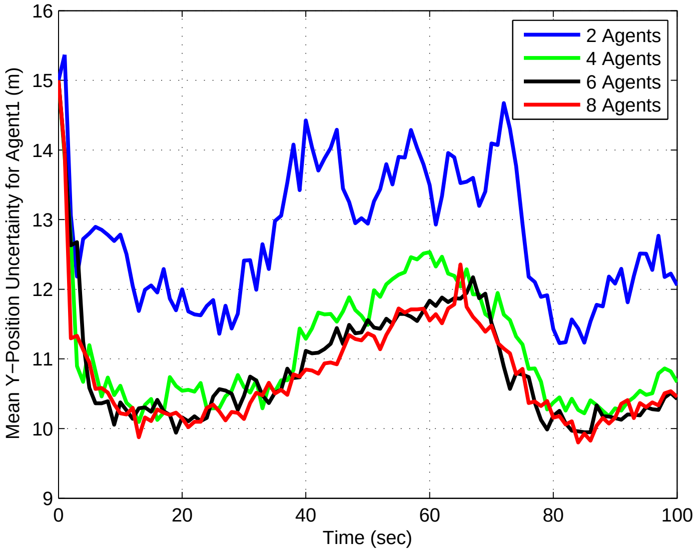


Figure 4.21: Agent1 y-position uncertainty analysis of estimation accuracy vs. number of agents (Over 100 Monte Carlo Trials)

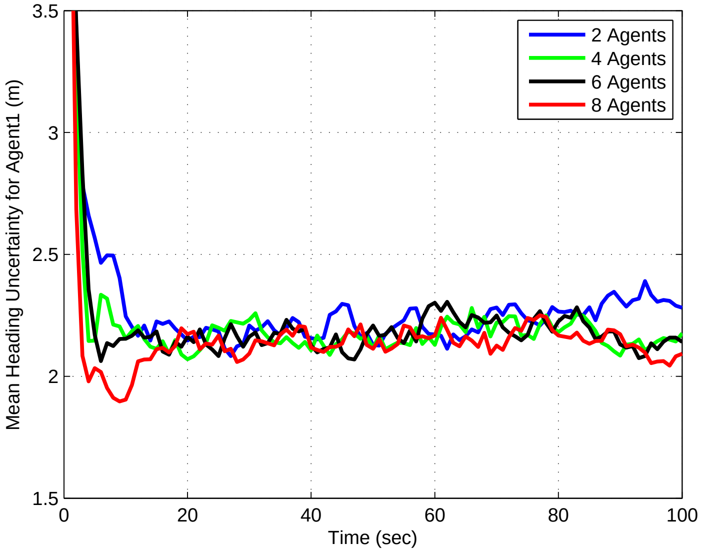


Figure 4.22: Agent1 heading uncertainty analysis of estimation accuracy vs. number of agents (Over 100 Monte Carlo Trials)

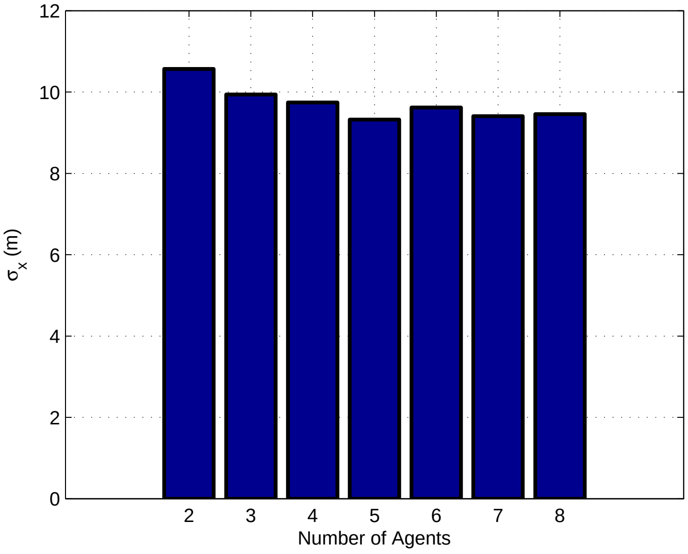


Figure 4.23: Bar plot showing the gradual improvement of Agent1's final x-position uncertainty as the number of agents is increased from 2 to 8 (Over 100 Monte Carlo Trials)

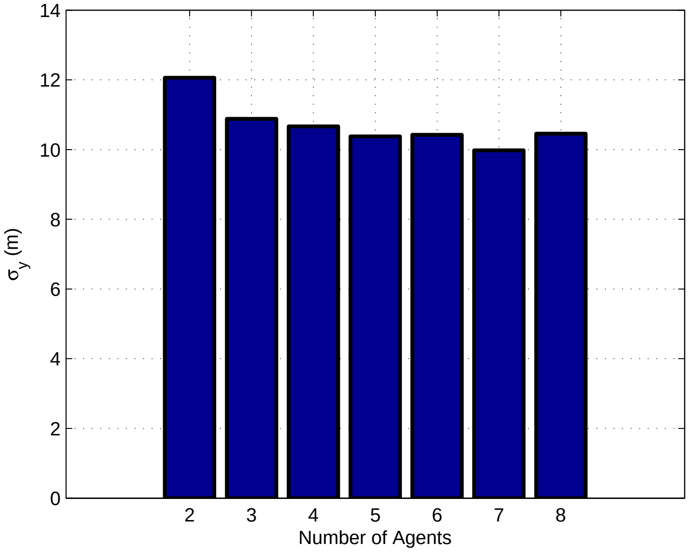


Figure 4.24: Bar plot showing the gradual improvement of Agent1's final y-position uncertainty as the number of agents is increased from 2 to 8 (Over 100 Monte Carlo Trials)

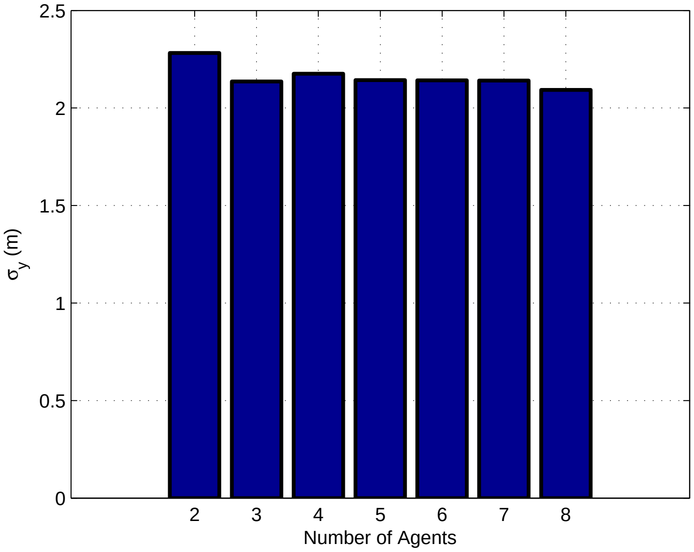


Figure 4.25: Bar plot showing the gradual improvement of Agent1's final heading uncertainty as the number of agents is increased from 2 to 8 (Over 100 Monte Carlo Trials)

of Shannon entropy $H(\mathbf{p}(\mathbf{x}))$, in the centralized processing scenario should be less than that obtained in the decentralized processing case. The Shannon entropy is defined as

$$H(\mathbf{p}(\mathbf{x})) = \sum_{i=1}^{N} \mathbf{p}(\mathbf{x}_i) \log\left(\mathbf{p}(\mathbf{x}_i)\right), \tag{4.16}$$

where the logarithm is considered to be the natural logarithm in this dissertation, $\mathbf{p}$ is a probability density function, and $\mathbf{x}_i$ represents the i^{th} sample from the sample set. Likewise, the Shannon Entropy obtained by the decentralized processing case should be less than the case where the agents are operating without any communication between then. The reduction in Shannon entropy from the no communication case to the decentralized processing case to the centralized processing case can be seen in Figure 4.26.

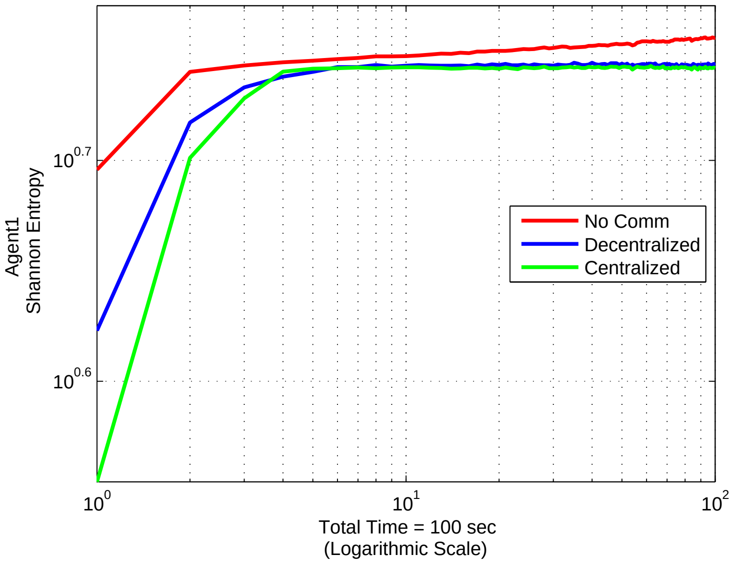


Figure 4.26: Shannon entropy for Agent1 in the centralized, decentralized, and no communication scenarios (Over 100 Monte Carlo Trials)

The same rationale in the Shannon entropy case also hold for the relative information case as can be seen in Figure 4.27 where the Kullback-Leibler divergence is calculated for the decentralized processing case and compared to the no communication case. In both architectures, the centralized case was used as the reference or target case.

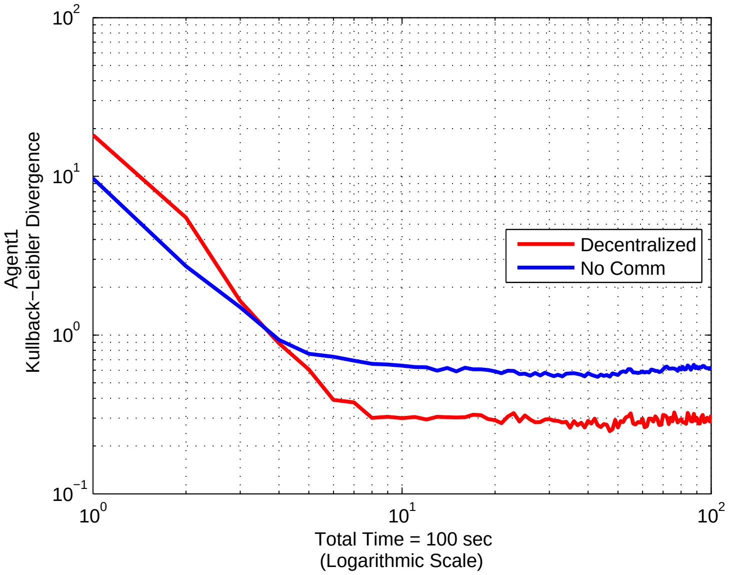


Figure 4.27: Kullback-Leibler divergence comparison between the decentralized and no communication scenarios for Agent1 with the centralized scenario used as the reference case (Over 100 Monte Carlo Trials)

4.6 Performance Under Various Measurement Scenarios

4.6.1 Range and Bearing Case. Most existing measurement configurations, in scenarios similar to the one considered here, incorporate both range and bearing measurements. However, there is an increasing body of literature concerned with bearings-only measurement configurations [216], [53], [272], [248]. Furthermore, there does exist a body of literature concerned with the range-only measurement case as well [105], [126]. The results shown for the following range and bearing measurement scenario, range only measurement scenario, and bearing only measurement scenario are the results obtained through 100 Monte Carlo runs with the parameters set to the values in Table 4.3.

In Figures 4.28, 4.29 and 4.30, agent1 specific state estimation errors are shown with units of meters, meters, and degrees for the agent's x-position, y-position, and heading angle respectively. Likewise, shown in the Figures 4.31 and 4.32 are agent1's landmark1 x and y position estimation errors. Note, only the estimation error and uncertainty bounds for the first landmark are given, since it is representative of the estimation error and un-

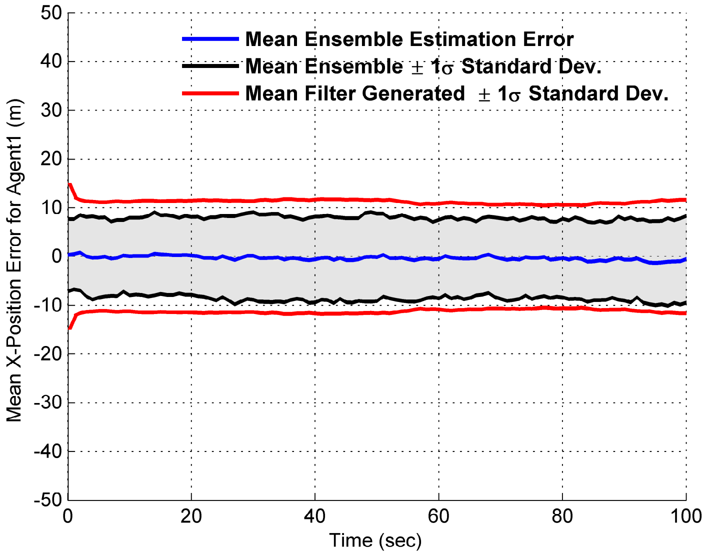


Figure 4.28: Agent1 x-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation bounds (red), and ensemble mean estimation error (blue) for the range and bearing measurement case (Over 100 Monte Carlo Runs)

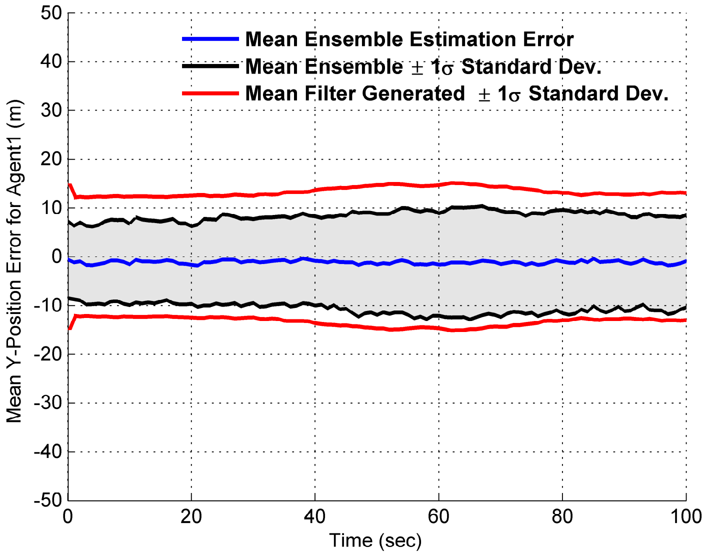


Figure 4.29: Agent1 y-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation bounds (red), and ensemble mean estimation error (blue) for the range and bearing measurement case (Over 100 Monte Carlo Runs)

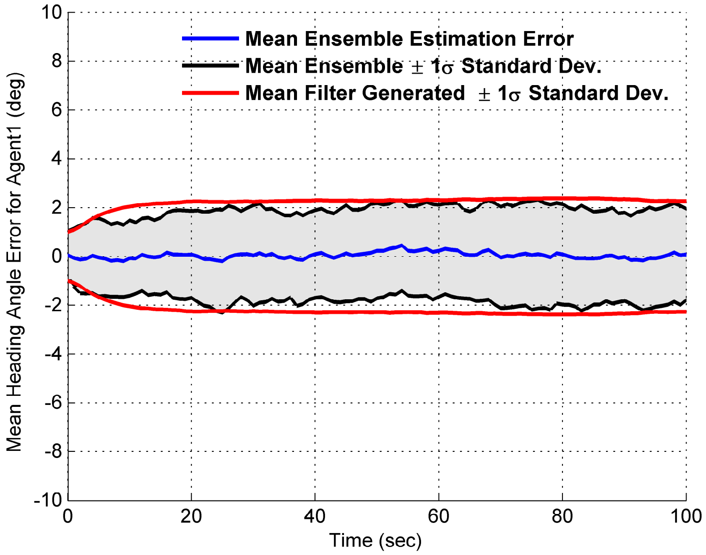


Figure 4.30: Agent1 heading angle state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation bounds (red), and ensemble mean estimation error (blue) for the range and bearing measurement case (Over 100 Monte Carlo Runs)

certainty for the two remaining landmarks. It can easily be seen that the estimation error resides well within the $\pm 1\sigma$ bounds. This trend could be representative of having over estimated the required amount of measurement noise strength. The implication of the filter performance shown is that additional tuning of the measurement noise intensities may be required.

4.6.2 Range Only Case.

This section is used to present the results of a range-only measurement scenario. In Figures 4.33 4.34, and 4.35 the estimation error and associated uncertainty are shown for agent1's vehicle specific states under the range only measurement scenario. In particular, notice how the estimation uncertainty for the heading state of agent1 in Figure 4.35 is larger than the estimation uncertainty obtained in the range and bearing measurement scenario. This should be expected, since there is no longer access to angular measurements. However, the lack of angular measurement doesn't imply that the estimation uncertainty will grow without bound, as indicated in Figures 4.33

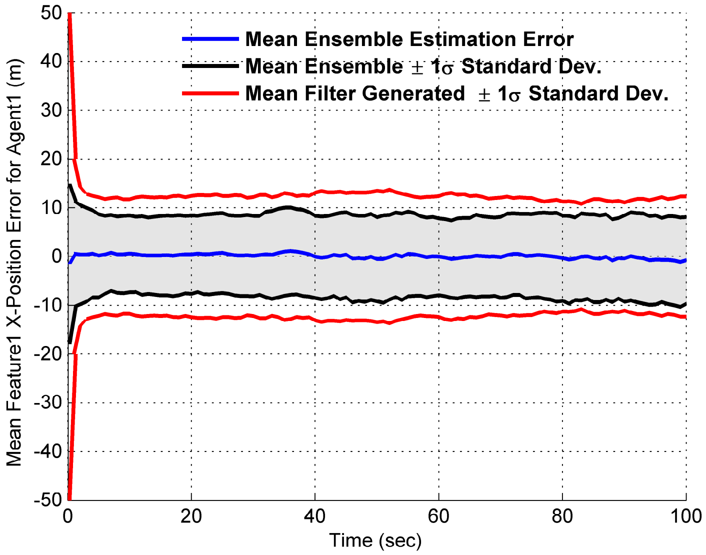


Figure 4.31: Agent1's landmark1 x-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation bounds (red), and ensemble mean estimation error (blue) for the range and bearing measurement case (Over 100 Monte Carlo Runs)

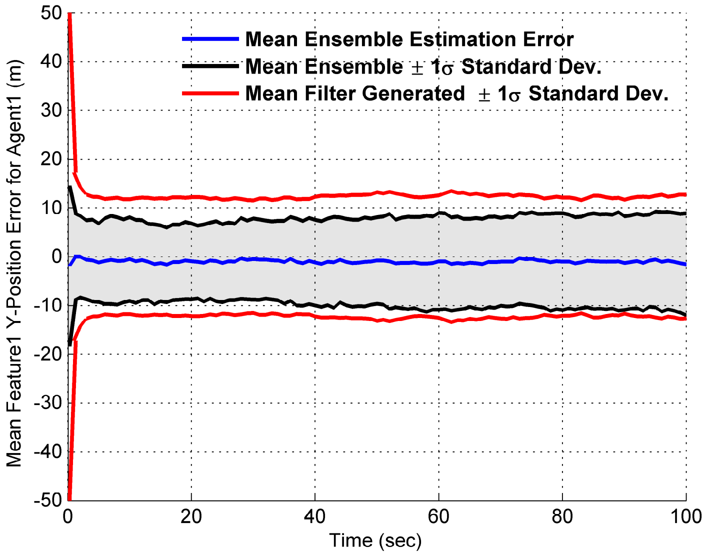


Figure 4.32: Agent1's landmark1 y-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation bounds (red), and ensemble mean estimation error (blue) for the range and bearing measurement case (Over 100 Monte Carlo Runs)

and 4.34. This is because the continued range measurement over filter iterations provides sufficient observability into the heading angle to retard the growth of the estimation error.

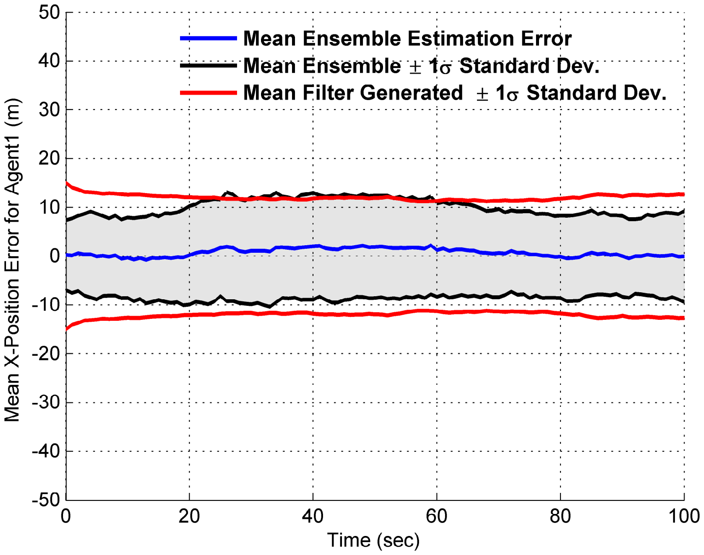


Figure 4.33: Agent1 x-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation bounds (red), and ensemble mean estimation error (blue) for the range only measurement case (Over 100 Monte Carlo Runs)

In Figures 4.36 and 4.37, the range only estimation error for agent1's landmark1 states are provided. As in the range and bearing case, the estimation error and corresponding uncertainty are given in units of meters. The uncertainty shown represents the $\pm 1\sigma$ bound.

4.6.3 Bearing Only Case. The final measurement scenario considered is the bearing-only measurement scenario, which is analogous to typical image-based navigation. In contrast to the range-only scenario in Section 4.6.2, notice that in Figures 4.38, 4.39, and 4.40 the estimation uncertainty for agent1 is larger in the position states and smaller in the heading states. The same logic used previously applies to the bearing-only measurement case as well. That is, access to a direct measurement of angle has the most

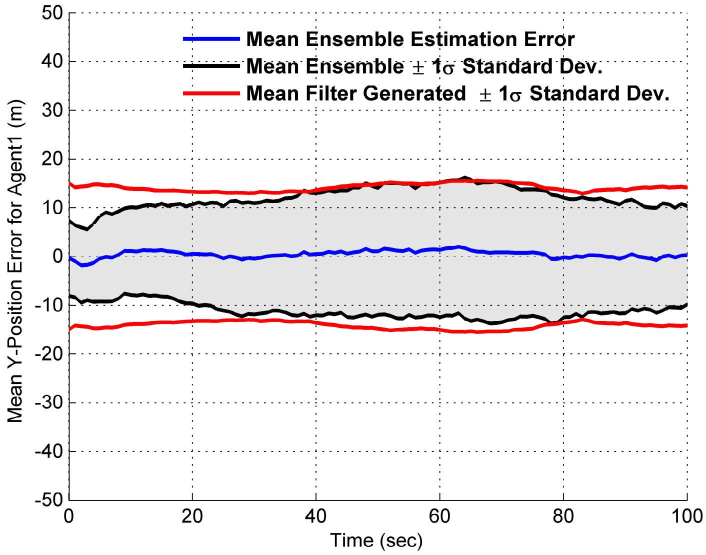


Figure 4.34: Agent1 y-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation bounds (red), and ensemble mean estimation error (blue) for the range only measurement case (Over 100 Monte Carlo Runs)

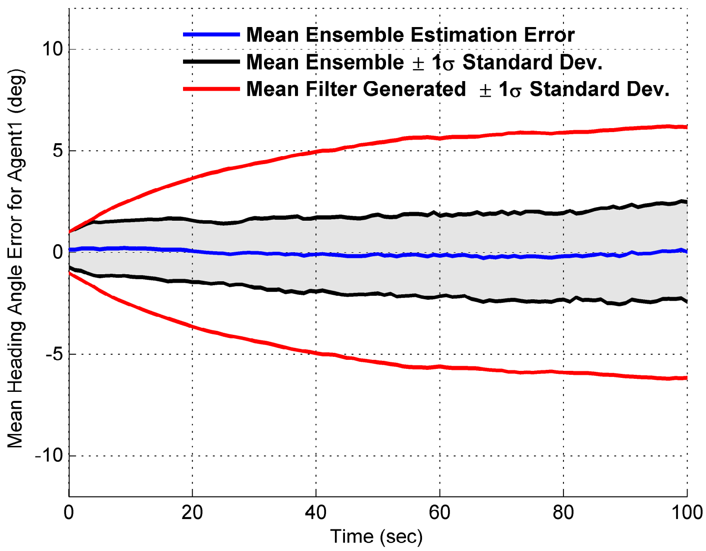


Figure 4.35: Agent1 heading angle state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation bounds (red), and ensemble mean estimation error (blue) for the range only measurement case (Over 100 Monte Carlo Runs)

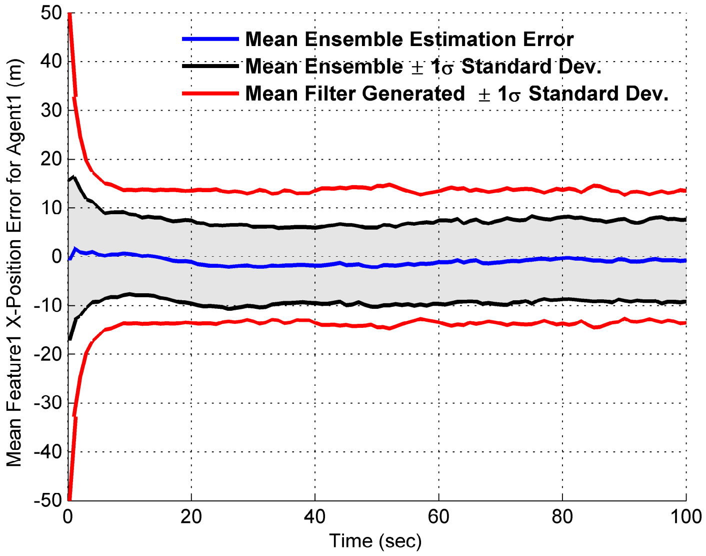


Figure 4.36: Agent1's landmark1 x-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation bounds (red), and ensemble mean estimation error (blue) for the range only measurement case (Over 100 Monte Carlo Runs)

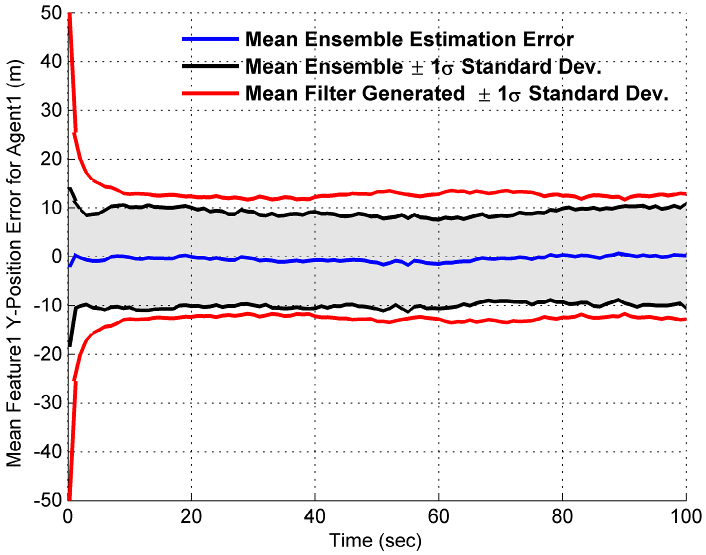


Figure 4.37: Agent1's landmark1 y-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation bounds (red), and ensemble mean estimation error (blue) for the range only measurement case (Over 100 Monte Carlo Runs)

impact on the angular state. In Figures 4.41 and 4.42, the bearing only landmark estimation errors for agent1's landmark1 state estimates are given in units of meters, along with $\pm 1\sigma$ bounds.

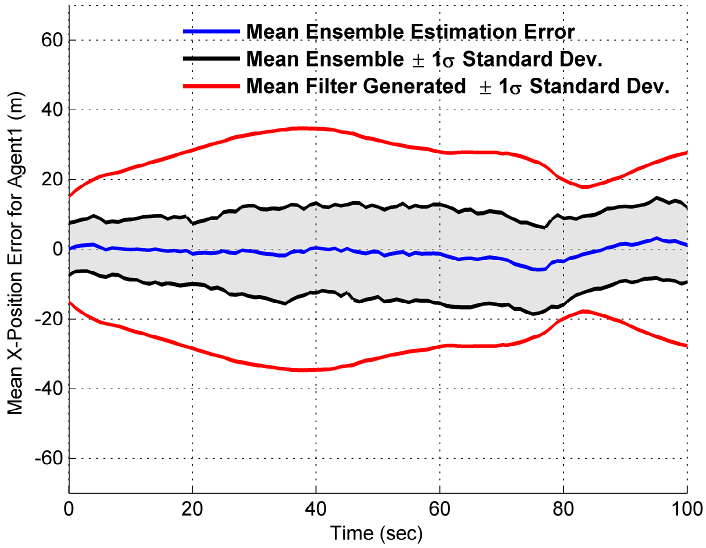


Figure 4.38: Agent1 x-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation bounds (red), and ensemble mean estimation error (blue) for the bearings only measurement case (Over 100 Monte Carlo Runs)

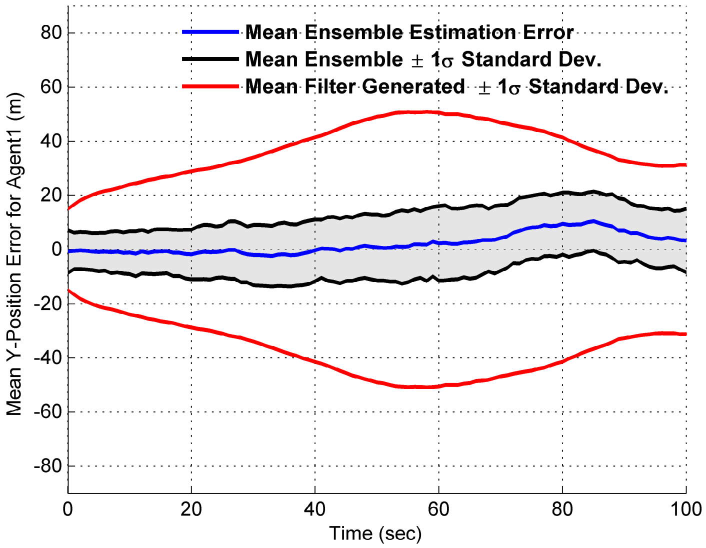


Figure 4.39: Agent1 y-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation bounds (red), and ensemble mean estimation error (blue) for the bearings only measurement case (Over 100 Monte Carlo Runs)

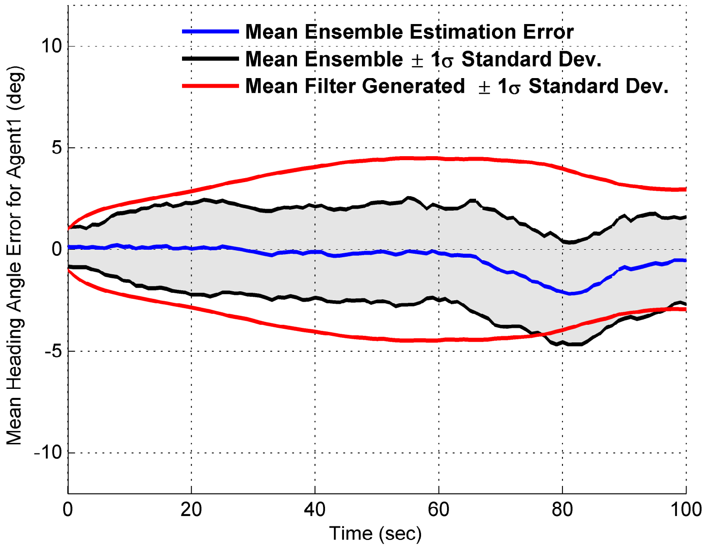


Figure 4.40: Agent1 heading angle state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation bounds (red), and ensemble mean estimation error (blue) for the bearings only measurement case (Over 100 Monte Carlo Runs)

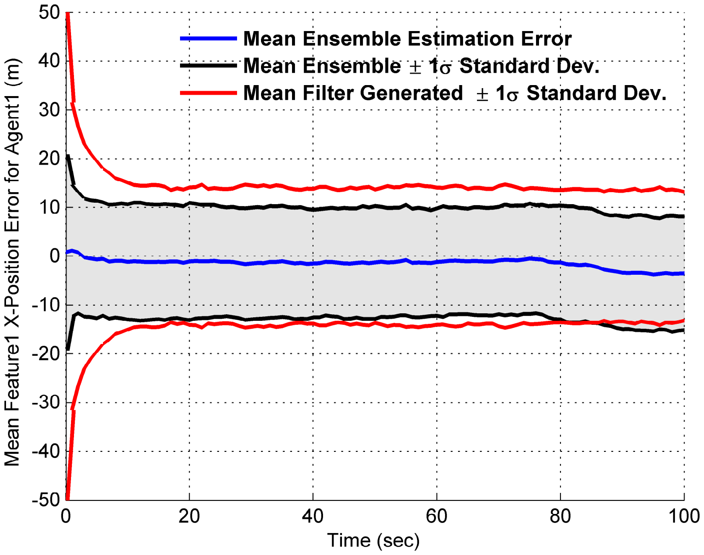


Figure 4.41: Agent1's landmark1 x-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation bounds (red), and ensemble mean estimation error (blue) for the bearings only measurement case (Over 100 Monte Carlo Runs)

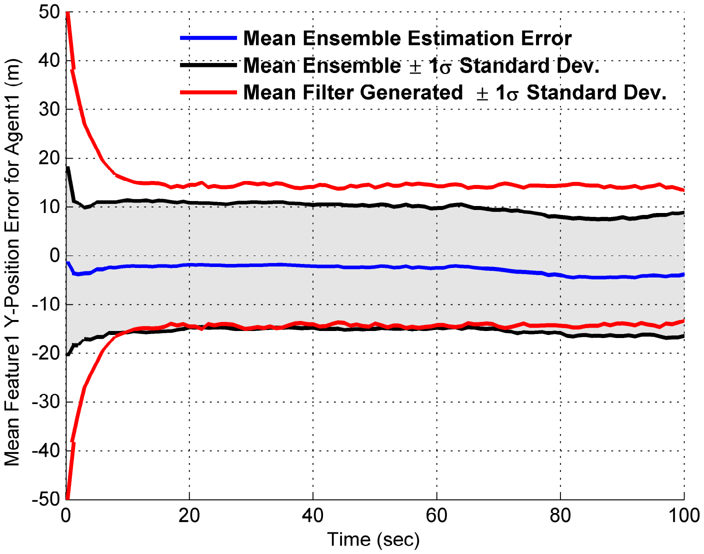


Figure 4.42: Agent1's landmark1 y-position state estimation errors with $\pm 1\sigma$ ensemble standard deviation bounds (black), $\pm 1\sigma$ mean filter generated standard deviation bounds (red), and ensemble mean estimation error (blue) for the bearings only measurement case (Over 100 Monte Carlo Runs)

4.7 State-of-the-Art Comparison

In this section, decentralized data fusion on the unit hypersphere is compared to current state-of-the-art methods for decentralized data fusion. In particular the methods used for comparison are the generalized covariance intersection method for Gaussian mixture models proposed by [287] and the traditional Covariance Intersection method.

Some interesting facts about the simulation results are worth noting here. First, the traditional Covariance Intersection method was not able to obtain meaningful results when the initial uncertainty was set to the values in Table 4.1. The reason for the poor performance of the Covariance Intersection approach can be attributed to the EKF formulation having to linearize about the current estimate, in addition to the nature of the resulting probability density function being inadequately described with Gaussian statistics.

The generalized GMM Covariance Intersection method was able produce reasonable results but required an excessive amount of computation time. The reason for the increased requirement for computation was primarily due to initialization with the *kmeans* clustering algorithm, coupled with the fact that the results of which were then used by the Expectation Maximization algorithm for defining the parameters for the GMM. Both the *kmeans* algorithm and the Expectation Maximization algorithms are iterative, and given poor initial conditions greatly impacts their convergence rate. Even though the estimation results obtained by the generalized GMM Covariance Intersection algorithm were certainly reasonable, it was certainly distinguishable from the fusion process on the unit hypersphere by the required amount of computation. In fact, the algorithm runtime was reported by MATLAB® to be approximately 433.9 seconds. In contrast, the runtime for data fusion on the unit hypersphere approach was reported to be approximately 20.4 seconds.

In an attempt to further exemplify the improvement in required computation time, we examined the run times between the two filters when the number of particles and the measurement types were changed. The results of the run time analysis are presented in Tables 4.2 and 4.3 for the GMMPF algorithm and the proposed decentralized Riemannian particle filter (DRPF) algorithm respectively. As the data clearly shows, an order of magnitude improvement in required computation time was achieved in all of the cases listed.

Table 4.2: PCCI mean run times varying measurements & particles

	1000 Particles	2500 Particles	5000 Particles	7500 Particles	10000 Particles
Range and Bearing	171.8 sec	283.4 sec	433.9 sec	633.6 sec	781.8 sec
Range Only	168.8 sec	279.8 sec	426.7 sec	629.3 sec	777.1 sec
Bearing Only	174.9 sec	287.1 sec	438.4 sec	638.4 sec	786.2 sec

Table 4.3: DRPF mean run times varying measurements & particles

	1000 Particles	2500 Particles	5000 Particles	7500 Particles	10000 Particles
Range and Bearing	9.3 sec	15.6 sec	20.4 sec	30.8 sec	39.6 sec
Range Only	9.8 sec	16.4 sec	20.7 sec	32.0 sec	41.2 sec
Bearing Only	9.9 sec	16.5 sec	21.5 sec	34.3 sec	42.8 sec

Now, a word of caution is in order. The comparison of runtime results between any algorithm collection should be viewed with a degree of skepticism. The results are subject to the available computer hardware, the degree of optimization of the relevant source code, and simulation environment used, among various other simulation parameters. However, given that the results shown were obtained on the same computer, within the same simulation environment, and by the authors own source code (with similar degrees of optimization), the runtime results presented do suggest at least an order of magnitude improvement in the required runtime. A direct comparison between the three algorithms in terms of mean filter generated 1σ standard deviations obtained by each of the three filter formulations can be seen in Figures 4.43, 4.44, and 4.45, which show the estimation uncertainty for agent1's x-position in meters, y-position in meters, and the heading in degrees. Recall

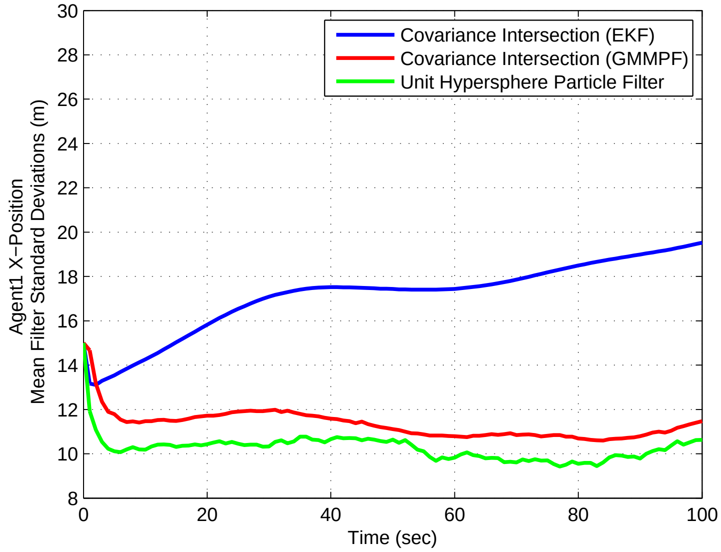


Figure 4.43: Agent1's x-position mean filter generated 1σ standard deviations comparison for the proposed information based unit hypersphere algorithm, traditional Covariance Intersection in an EKF framework, and Gaussian Mixture Model Particle Filter (GMMPF) (Over 100 Monte Carlo Runs)

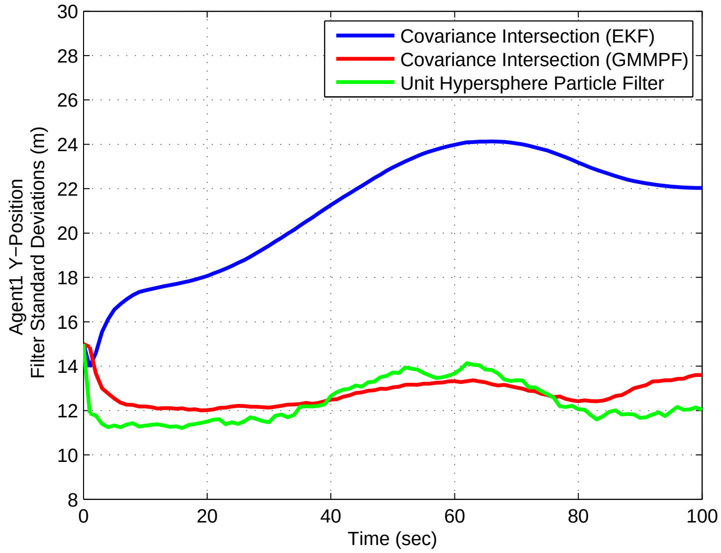


Figure 4.44: Agent1's y-position mean filter generated 1σ standard deviations comparison for the proposed information based unit hypersphere algorithm, traditional Covariance Intersection in an EKF framework, and Gaussian Mixture Model Particle Filter (GMMPF) (Over 100 Monte Carlo Runs)

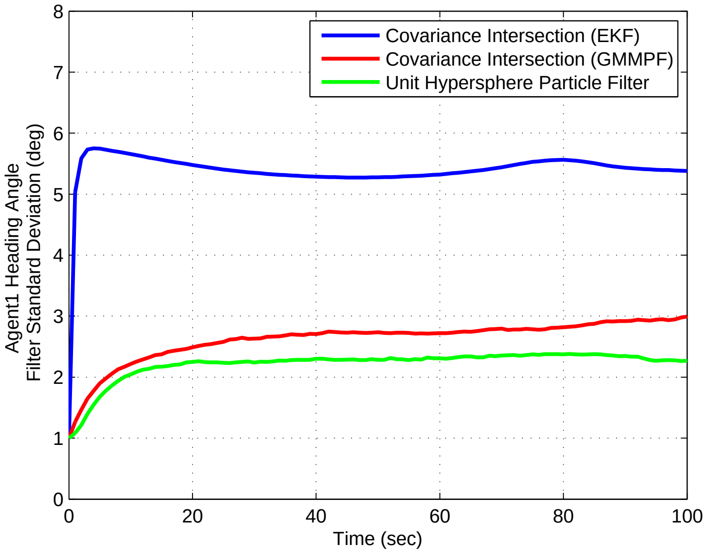


Figure 4.45: Agent1's heading angle mean filter generated 1σ standard deviations comparison for the proposed information based unit hypersphere algorithm, traditional Covariance Intersection in an EKF framework, and Gaussian Mixture Model Particle Filter (GMMPF) (Over 100 Monte Carlo Runs)

that back in Section 4.4.2, the resulting probability density functions that were the result of the unit hypersphere fusion were shown to be adequately described as being Gaussian in nature. This fact would seem to contradict the claim of poor Gaussian descriptions as being a cause for the inferior estimation performance of the traditional Covariance Intersection approach. However, recall that the Gaussian description was appropriate for the landmark states, since they were what was being communicated between the agents. The Gaussian description did not pertain to the probability density functions that described the agent specific states. The Gaussian description is inadequate for the agent states as a result of the choice of models used in the simulation. The process model describing the agent states had notable nonlinearities, as can be seen in Equation (4.8). Where the process model for the landmark states was completely linear due to their being modeled as stationary landmarks.

4.8 Summary

This chapter was used to present the simulation environment and results. Both process and measurements models were detailed. Assumptions used in the simulation and models were given explicitly. The analysis was divided into four distinct categories.

The first category focused on comparing the two proposed algorithms. The algorithms were shown to produce similar estimation results.

The second category was concerned with the operational integrity of the proposed algorithm. The algorithms integrity was evaluated against various conditions to include a rudimentary consistency analysis, communication topologies, number of agents, number of particles, and finally in terms of Shannon entropy and Kullback-Leibler divergence.

The third category was comprised of several different measurement scenarios used to validate the proposed algorithms of Chapter III. Scenarios included range and bearing measurements, range only measurement, and bearing only measurement scenarios.

The fourth category was dedicated to comparing the derived algorithms with currently available methods in the decentralized data fusion literature. The proposed algorithms out performed both the traditional Covariance Intersection algorithm formulated in an extended Kalman filter approach, as well as, the Covariance Intersection approach to Gaussian Mixture Model particle filtering. The GMM particle filtering approach was able to produce comparable estimation results, but with more than an order of magnitude increase in the required computation time. The next chapter will provide a summary of the research performed, highlight the proposed research contributions, and identify areas worthy of further research.

V. Conclusions

5.1 Introduction

This chapter is intended to serve multiple purposes, chief among them is to delineate between the research performed and the proposed scientific contributions stated in Chapter I. Another purpose of this chapter is to draw conclusions based on the presented work. Furthermore, this chapter will offer potential avenues worthy of future research that were identified throughout the course of this research effort.

The body of work presented in this dissertation has extended the current state of the art in decentralized particle filtering. The general research field of data fusion is vast and offers several interesting research questions. The subclass of problems concerned with decentralized data fusion is no exception. One of the questions addressed in this research was the formulation of decentralized particle filtering algorithms capable of producing estimates that did not suffer from the incorporation of redundant information, also known as the data incest problem or inconsistent fusion problem. If the problem of inconsistent estimation is not addressed appropriately, it will likely lead to overly optimistic estimates, and eventual filter divergence. There have been several approaches to solving the inconsistent estimation problem offered throughout the available literature. The novel approaches offered in this dissertation exploited the synergetic relationship that has been shown to exist between differential geometry and nonlinear filtering.

The fact that geometry and filtering are intimately tied is not a secret. In fact, a geometric methodology was used by Kalman in his original derivation of the now widely used Kalman filter [148]. Even the state of the art solution methods to the inconsistent fusion problem, like Covariance Intersection, look to exploit the geometric relationships between covariance matrices to formulate convex optimization problems.

5.2 Value Added by the Research Effort

This research effort has developed novel decentralized particle filtering algorithms based on the correspondences that exist between the research fields of differential geometry and nonlinear filtering. The well understood differential geometry of the unit hypersphere played a pivotal role in the formulation of the decentralized particle filtering algorithms presented in Chapter III.

A key research contribution was made through the demonstration of a never before used general framework for performing decentralized particle filtering that is based on a non-Euclidean geometric interpretation of decentralized data fusion. Current decentralized filtering methods represent a dichotomy of techniques. The first class of methods requires the ability to linearize models so that Kalman based methods can be used for decentralized data fusion. The second class of methods makes use of particle filtering technology by requiring that complex filtering densities be represented with mixture models for decentralized data fusion. Our framework relies on no such requirements.

Another research contribution was made by projecting probability density functions onto the surface of the unit hypersphere, mainly that filtering calculations were now able to be performed in closed-form. The use of closed-form calculations has impacted the field of decentralized particle filtering in primarily two ways. First, by no longer requiring costly iterative numerical approximations to filtering operations, the implementation of algorithms that require significantly less computational resources are made available, which ultimately improves computational efficiency. Second, the proposed algorithms removed the implementation bottleneck of having to perform computationally costly parameterization procedures associated with the conversion of particle representations into continuous probability density representations, and in so doing, does not constrain the type of probability density functions being considered.

Additional research contributions can be seen through the rigorous analysis of results obtained in simulation, wherein the proposed decentralized particle filtering algorithms were shown to provide superior fusion performance over currently available algorithms, under a variety of scenarios in under a variety of performance metrics. For example, analysis was performed based on the impact of the number of particles and the number of agents used on the estimation performance. The information and entropy content of simulation results were examined, along with various measurement scenarios. Furthermore, we successfully adapted an algorithm capable of providing existence and uniqueness guarantees for solutions to the decentralized particle filtering problem under mild assumptions. Existence and uniqueness guarantees are not associated with existing approaches unless under restrictive assumptions to the network topology or available probabilistic representations.

Although, it was shown that the performance gains with respect to achievable accuracy were modest when compared to the popular GMM particle filtering formulation. However, the true value added when compared to the GMM particle filtering formulation can be seen in the reduction in required algorithm runtime of an order of magnitude in the scenario studied.

The following is a summary list of the proposed novel research contributions of the work documented in this dissertation, and can also be found in Chapter I.

1. Demonstrated a never before used general framework for performing decentralized particle filtering based on a non-Euclidean geometric interpretation.
2. Presented decentralized particle filtering algorithms that provide closed form filtering calculations, currently unavailable in the general case.
3. Adapted an algorithm capable of providing existence and uniqueness guarantees for solutions to the decentralized particle filtering problem.
4. Established a technology bridge between multiple research communities that permits access to previously unavailable analysis tools.
5. Demonstrated, through empirical evidence, that an order of magnitude improvement in computational performance is possible with the proposed algorithms.

In the opinion of this author, the research presented in this dissertation along with the existing literature on the use of differential geometric methods in nonlinear filtering applications have only begun to scratch the surface of possible formulations. Throughout the course of this research, several interesting research problems and hints to potential formulations of solution approaches have manifested themselves in one form or another. It is doubtful that any single researcher can provide the necessary attention to all of the identified research questions; however, a few of the more prominent ones are mentioned next.

5.3 Areas Worthy of Future Considerations

The research conducted in this dissertation has highlighted multiple areas worthy of further research. The following is a partial list of interesting research questions left open.

1. More realistic environmental modeling to include non-point feature representations, non-stationary feature models, full six degree of freedom kinematics, *etc.*
2. More realistic sensor models in the form of investigating the impact of limited sensor range, limited field of view (FOV), *etc.*
3. The incorporation of a decision maker in the algorithm for purposes of path planning, target assignment, *etc.*
4. Investigate the feasibility of formulating the entire fusion process without having to ever leave the unit hypersphere.
5. Investigate the impact of higher fidelity density estimation techniques on algorithm performance.
6. Investigate the utility of other differential geometric surfaces for use in the decentralized data fusion process.
7. A detailed analysis of the impact of limited communications or intermittent communications.

8. Investigate the utility of using the unit hypersphere in defining a potential *rule-of-thumb* for establishing necessary sample size.
9. Explore how additional tasks associated with navigation and tracking be formulated in this framework *eg.*, data association, feature detection, map management, *etc.*
10. Investigate the relationships between the geometry of the agents, the available sensors on individual agents, and achievable estimation accuracy of the agents.
11. Integrate so called *down stream* functions like guidance and control to investigate the closed-loop performance.
12. Identify metrics or guidelines for determining when agents should communicate and when communication may be counter-productive?
13. Investigate techniques for describing the information that agents share about themselves and the environment? Furthermore, determine if the information description is universal or situationally dependent.
14. Finally, investigate methods for monitoring network health, identifying misinformation and malicious network attacks, and methods to remedy network intrusions.

A deeper understanding of the fundamental role that information plays in decentralized data fusion has yet been adequately explored. In the opinion of this author, information and its interpretation are at the core of truly understanding multi-agent systems. The determination of how information percolates throughout a multi-agent system is an interesting research agenda worthy of consideration.

Clearly, great benefit resides in the implementation of decentralized data fusion. The true benefit is still somewhat hazy. It will likely become more focused as some of the questions listed above begin to be answered.

5.4 Final Thoughts

The true power of the differential geometric framework originates from its ability to accommodate an assortment of scenarios. For example, in the event that there exists the need to compare two probability density functions as is the case in filtering problems, signal detection problems, and feature classification problems, the framework can be utilized. In the case of redundant information, the intuitive representation of the abstract concepts of similarity and information were made possible through the use of differential geometry.

The rate at which hardware and software technologies are maturing is making decentralized data fusion methods a practical option for many application areas. From an Air Force perspective, applications such as navigation, tracking, and targeting can benefit from decentralization. Likewise, Air Force applications impose considerable constraints on timing, reliability, and accuracy. The ability to provide timely and reliable information to decision makers is vital to the success of the Air Force mission. This research has shown through the use of differential geometry that timeliness and reliability are realizable when employing decentralized systems.

The research has advanced the state of the art in decentralized data fusion. However, there is still significant work to be done if systems based on a decentralized architecture are to be realized in an Air Force operational environment.

Appendix A. Topics From Topology and Real Analysis

A.1 Introduction

This appendix is used to provide background definitions for topics typically found in topology and analysis. The reason is twofold. First, in order to benefit from the use of manifolds, one needs to assume some basic topological structure is available. Second, to serve as an accessible presentation in the event that a reader is unfamiliar with topology and/or analysis. The material found in this appendix can be found in a number of excellent references. The primary sources for this appendix include textbooks [69], [97], [202], [14], [245], [119], [204] and lecture notes obtained during the following courses [85], [215], [86], [87].

A.2 Definitions From Analysis

Definition A.2.1. *A **collection** is used to refer to a set of objects whose elements are also sets.*

Definition A.2.2. *A **partition** of a set A is a collection of disjoint nonempty subsets of A whose union is all of A.*

Definition A.2.3. *If set X possess an order relation, say $<$, and if $a < b$ then (a, b) represents the set*

$$\{x \mid a < x < b\}; \tag{A.1}$$

*and is called an **open interval** in X. If X is empty a is the **immediate predecessor** of b, and b is the **immediate successor** of a.*

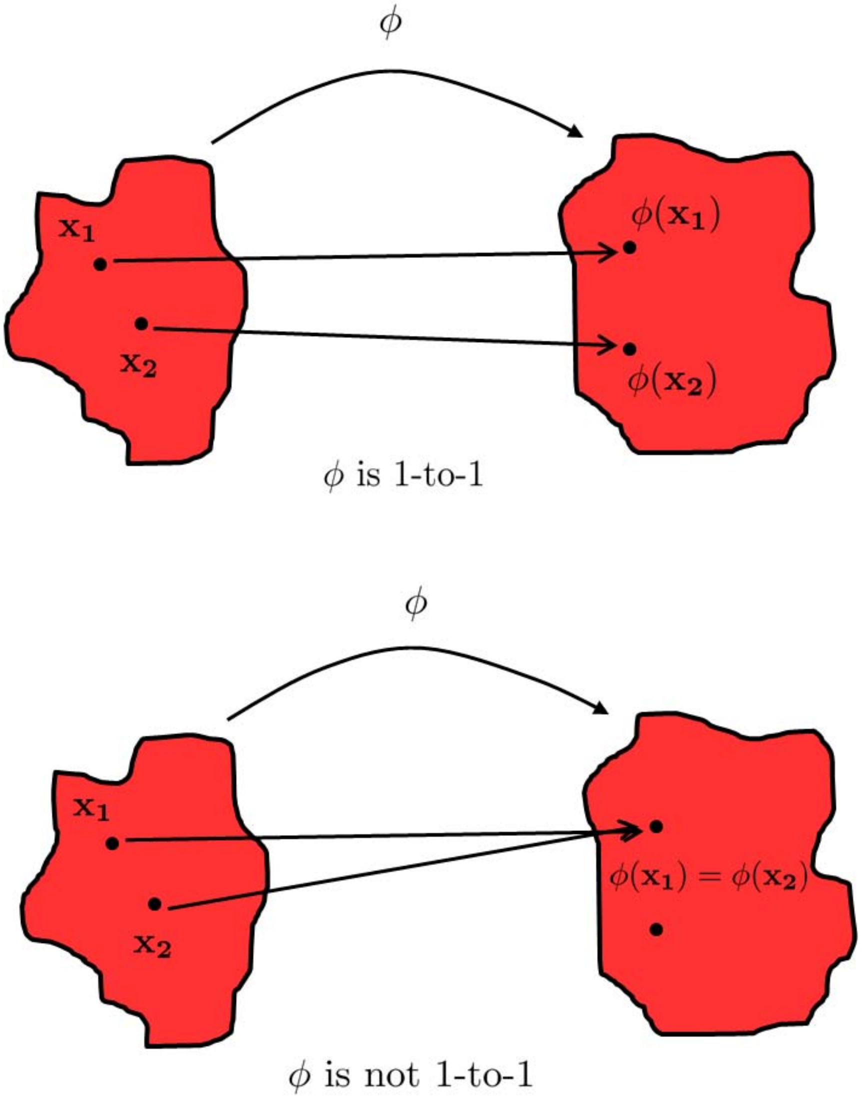
Definition A.2.4. *A function $\phi : A \to B$ is said to be **injective** or **one-to-one** if for every pair of distinct points in A, their image under ϕ are distinct.*

$$[\phi(a) = \phi(a')] \Rightarrow [a = a'] \tag{A.2}$$

Definition A.2.5. *A function $\phi : A \to B$ is said to be **surjective** or **onto** if every element of B is the image of some element of A under the function ϕ.*

$$[b \in B] \Rightarrow [b = \phi(a), \quad \textit{for at least one} \quad a \in A] \tag{A.3}$$

Definition A.2.6. *A function $\phi : A \to B$ is said to be **bijective** if it is both injective and surjective.*



ϕ is 1-to-1

ϕ is not 1-to-1

Figure A.1: An injective mapping. The original figure can be found in reference [202].

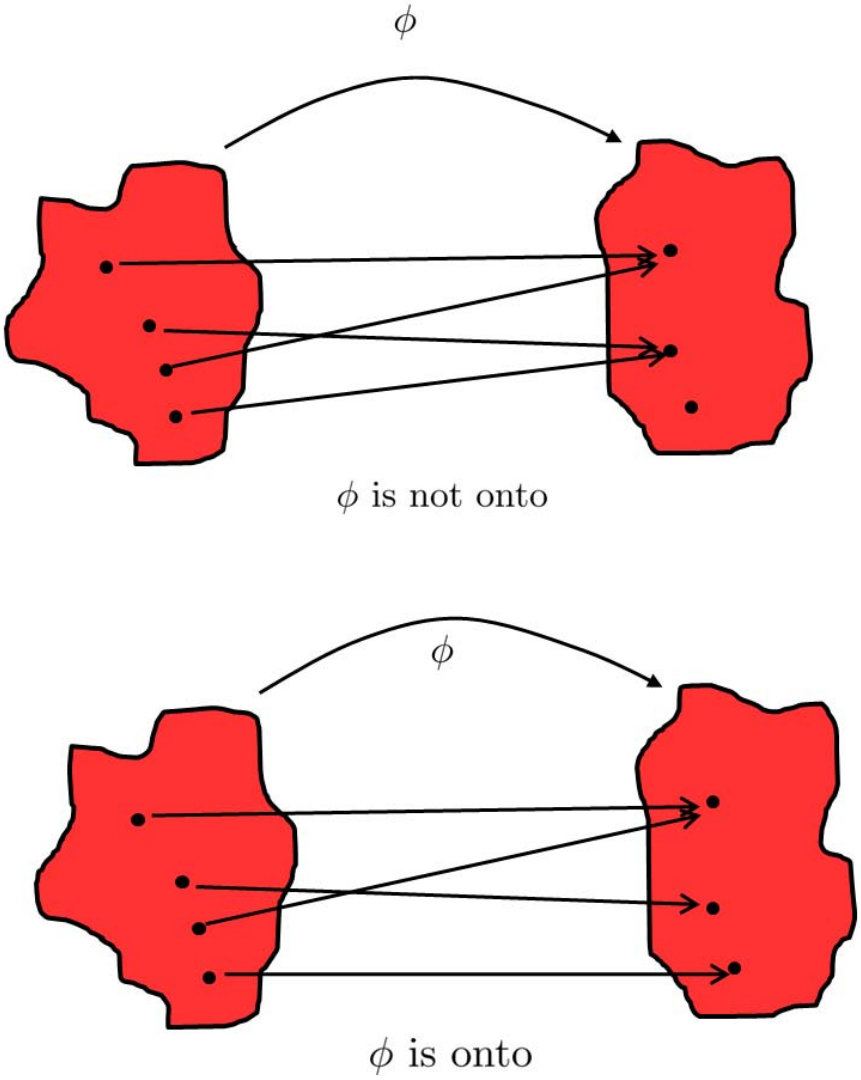


Figure A.2: A surjective mapping. The original figure can be found in reference [202].

Definition A.2.7. *Let b be an element of a subset of A called A_0. If $x \leqslant b$ for every x in A_0 then A_0 is **bounded** above and b is called an **upper bound**.*

Definition A.2.8. *If the set of all upper bounds for A_0 has a smallest element, that element is called the **supremum** or **least upper bound***

Definition A.2.9. *Let b be an element of a subset of A called A_0. If $x \geqslant b$ for every x in A_0 then A_0 is **bounded below** and b is called a **lower bound**.*

Definition A.2.10. *If the set of all lower bounds for A_0 has a largest element, that element is called the **infimum** or **greatest lower bound***

A.3 Definitions From Topology

Definition A.3.1. *A set A is said to be **countably infinite** if there exists a bijective correspondence*

$$f : A \to \mathcal{Z}_+, \quad (\mathcal{Z} \text{ is the integers})$$

Definition A.3.2. *A **topology** on a set X is a collection of subsets $\mathcal{T}$ of X having the following properties:*

1. *$\emptyset$ and X are in $\mathcal{T}$*
2. *The union of any of the elements of any subcollection $\mathcal{T}$ is in $\mathcal{T}$*
3. *The intersection of the elements of any finite subcollection of $\mathcal{T}$ is in $\mathcal{T}$*

*A set X that has a defined topology is called a **topological space**.*

Definition A.3.3. *A topological space X is called a **Hausdorff space** if for every pair of distinct points x_1 and x_2 there exist neighborhoods U_1 and U_2 of x_1 and x_2 respectively, that are disjoint.*

Definition A.3.4. *Let X and Y be topological spaces; let $\phi : X \to Y$ be a bijection. If the function ϕ and its inverse ϕ^{-1} are continuous, then ϕ is called a **homeomorphism**.*

Definition A.3.5. *Let U and V be open sets in $\mathbb{R}^n$. A homeomorphism $\phi : U \to V$ from U onto V is called a C^∞ differentiable homeomorphism or **diffeomorphism**, if both ϕ and ϕ^{-1} are continuous and infinitely differentiable C^∞.*

A.4 Norms, Metrics, and Inner Products

Definition A.4.1. *Let V be a set endowed with operations of addition and scalar multiplication. If the elements of V are real valued then V is called a **real vector space** if the following conditions are satisfied for all $x, y, z \in V$, and all scalars $c, c_1, c_2 \in \mathbb{R}$:*

1. $x + y = y + x$

2. *$x + (y + z) = (x + y) + z$*
3. *There exists a set called the empty set denoted as $\emptyset$ such that $x + \emptyset = x$*
4. *For all elements x there exists a unique element $-x$ such that $x + (-x) = 0$*
5. *$1 \cdot x = x$*
6. *$(c_1 \cdot c_2) \cdot x = c_1 \cdot (c_2 \cdot x)$*
7. *$c \cdot (x + y) = c \cdot x + c \cdot y$*
8. *$(c_1 + c_2) \cdot x = c_1 \cdot x + c_2 \cdot x$*

Definition A.4.2. *A **subspace** V_0 of a vector space V is a nonempty subset of V which satisfies the following two requirements:*

1. *For any pair $x, y \in V_0$, $x + y \in V_0$*
2. *For any x in V_0 and any scalar c, $c \cdot x \in V_0$*

Definition A.4.3. *An **inner product** on a real vector space V is a real function denoted as*

$$\langle x, y \rangle : V \times V \to \mathbb{R} \tag{A.4}$$

such that for all $x, y, z \in V$ and all $c \in \mathbb{R}$ the following is true:

1. *$\langle x, y \rangle = \langle y, x \rangle$*
2. *$\langle c \cdot x, y \rangle = c \cdot \langle x, y \rangle$*
3. *$\langle x+, z \cdot y \rangle = \langle x, z \rangle + \langle y, z \rangle$*
4. *$\langle x, y \rangle > 0, \quad \forall x \neq 0$*

Definition A.4.4. *A **norm** on a real vector space V is a mapping such that*

$$V = \mathbb{R}^n \to \mathbb{R} \tag{A.5}$$

and is typically denoted by

$$\| \cdot \| : V \to [0, \infty) \tag{A.6}$$

*such that for all $x, y \in V$ and scalars c the definition for vector space, subspace, and inner product hold. A vector space endowed with a norm is called a **normed space**.*

Remark 1. *A norm $\| \cdot \|$ defines a metric $d(x, y) = \|x - y\|$ on V, i.e., a function that measures the* distance *between two elements x and y of V, such that the following four properties hold $\forall x, y, z \in V$*

1. **Symmetry:** $d(x, y) = d(y, x)$
2. **Positive Definite:** $d(x, y) \geqslant 0, \forall x \neq y$

3. **Equality Condition:** $d(x, x) = 0$
4. **Triangle Inequality:** $d(x, z) \leqslant d(x, y) + d(y, z)$

Definition A.4.5. *A **metric** on a vector space V is a mapping (function)*

$$d(\cdot, \cdot) : V \times V \to [0, \infty) \tag{A.7}$$

*that satisfies the properties of a norm $\forall x, y, z \in V$. A Space endowed with a metric is called a **metric space**.*

Remark 2. *In this definition of a metric space, the space V is not necessarily a vector space. In fact, any space endowed with a metric is a metric space.*

Remark 3. *Inner products and norms may not be defined on metric spaces!*

A.5 Hilbert and Banach Spaces

Definition A.5.1. *A **Hilbert space** H is a vector space endowed with an inner product and associated norm and metric, such that every Cauchy sequence in H has a limit in H.*

Remark 4. *Consider a Euclidean space $\mathbb{R}^n$. It is obviously a vector space endowed with the usual inner product, norm, and associated metric given by,*

1. **Inner Product:** $\langle x, y\rangle = x^T y$
2. **Norm:** $\|x\| = \sqrt{x^T x} = \sqrt{\langle x, x\rangle}$
3. **Metric:** $\|x - y\|$

such that every Cauchy sequence takes a limit in $\mathbb{R}^n$. This makes $\mathbb{R}^n$ a Hilbert space.

Definition A.5.2. *A **Banach space** B is a normed space with associated metric*

$$d(x, y) = \|x - y\| \tag{A.8}$$

such that every Cauchy sequence in B has a limit in B.

Remark 5. *The difference between a Banach space and a Hilbert space is the source of the norm. In Hilbert spaces, the norm is defined via the inner product and in Banach spaces the norm is defined directly from the definition.*

Bibliography

1. A. Kong, W.H. Wong, J.S. Liu. “Sequential Imputations and Bayesian Missing Data Problems”. *Journal of American Statistical Association*, 89:278 – 288, 1994.

2. A. Pahliani and P. Lima. “Cooperative opinion pool: a new method for sensor fusion by a robot team”. *Intelligent Robots and Systems*, 2007.

3. Abdelkrim Nemra and Nabil Aouf. “Robust Airborne 3D Visual Simultaneous Localization and Mapping with Observability and Consistency Analysis”. *Journal of Intelligent Robot Systems*, 55:345–376, January 2009.

4. Abhijit Sinha, Huimin Chen, D.G. Danu, Thia Kirubarajan, M. Farooq. “Estimation and Decision Fusion: A Survey”. *Neurocomputing*, 71:2650–2656, 2008.

5. Adrian Peter and Anand Rangarajan. “Information Geometry for Landmark Shape Analysis: Unifying Shape Representation and Deformation”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):337–350, Feb. 2009.

6. A.G. Dabak. *A Geometry for Detection Theory*. Phd thesis, Rice University, Department of Electrical and Computer Engineering, 1992.

7. Akashi, Kumamoto H., H. “Random Sampling Approach to state estimation in Switching Environments”. *Automatica*, 13:429–434, 1977.

8. Ali, S. M. and S. D. Silvey. “A General Class of Coefficients of Divergence of One Distribution From Another”. *J. Royal Statistical Society*, 28:Series B, 1966.

9. Alvina Goh and Rene Vidal. “Clustering and Dimensionality Reduction on Riemannian Manifolds”. *IEEE Conference on Computer Vision and Pattern Recognition*, 1–7. Anchorage, AK, 23-28 June 2008.

10. S. l. Amari, R. E. Kass S. L. Lauritzen, O. E. Barndorff-Nielsen and C. R. Rao. *Differential Geometry in Statistical Inference*, volume 10. Institute of Mathematical Statistics Lecture Notes-Monograph Series, 1987.

11. Amari, S. “Information Geometry of statistical inference - an overview”. *Proceedings of 2002 IEEE Information Theory Workshop, Bangalore*, 86–89. 2002.

12. Amari, S. and H. Nagaoka. *Methods of information geometry*. American Mathematical Society, New York, 1993.

13. Anbang Yao, Xinggang Lina, Guijin Wanga and Xiujuan Chaib. “An Incremental Bhattacharyya Dissimilarity Measure for Particle Filtering”. *Pattern Recognition*, 43(4):1244–1256, April 2010.

14. Apostol, T.M. *Mathematical Analysis*. Addison Wesley, 1974.

15. A.R. Benaskeur. “Consistent fusion of correlated data sources”. *IEEE 28th Annual Conference of the Industrial Electronics Society*, volume 4, 2652 – 2656. 5-8 Nov 2002.

16. Arnaud Doucet, Neil Gordan, Nando de Freitas. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

17. Arthur Gelb, Raymond A. Nash Jr. Charles F. Price Arthur A. Sutherland Jr., Joseph F. Kasper Jr. *Applied Optimal Estimation*. The M.I.T. Press, 1974.

18. Arturo Gil, Mónica Ballestaa, Óscar Reinosoa and Miguel Juliá. “Multi-Robot Visual SLAM Using a Rao-Blackwellized Particle Filter”. *Robotics and Autonomous Systems*, 58(1):68–80, January 2010.

19. Arulampalam, Sanjeev, Simon Maskell, and Neil Gordon. “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking”. *IEEE Transactions on Signal Processing*, 50:174 – 188, 2002.

20. Arulampalam, Sanjeev, Simon Maskell, Neil Gordon, and Tim Clapp. “A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking”. *IEEE Transactions on Signal Processing*, 50:174–188, 2001. URL `http://www.cs.ubc.ca/\~{}murphyk/Software/Kalman/ParticleFilterTutorial.pdf`.

21. Azimi-Sadjadi, B. *Approximate Nonlinear Filtering With Applications to Navigation*. Ph.d. dissertation, University of Maryland at College Park, College Park, MD USA, 20742, August 2001.

22. Azimi-Sadjadi, B. and P.S. Krishnaprasad. “Approximate nonlinear filtering and its applications for GPS”. *Proceedings of the 39th IEEE Conference on Decision and Control*, (August):1579–1584, 2000. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=912085`.

23. Azimi-Sadjadi, B. and PS Krishnaprasad. “Integer ambiguity resolution in GPS using particle filtering”. *2001 American Control Conference, Arlington, VA*, 3761âĂŞ3766. 2001. URL `http://www.csa.com/partners/viewrecord.php?requester=gs\&amp;collection=TRD\&amp;recid=A0145892AH`.

24. Azimi-Sadjadi, B. and P.S. Krishnaprasad. “A Particle Filtering Approach to Change Detection for Nonlinear Systems”. *EURASIP Journal on Applied Signal Processing*, 2004:2295–2305, 2004.

25. Azimi-Sadjadi, P.S., B. Krishnaprasad. “Change Detection for Nonlinear Systems; A Particle Filtering Approach”. *Proceedings of the American Control Conference*, volume 5, 4074 – 4079. 2002.

26. B. F. La Scala, M. Mallick, S. Arulampalam. “Differential Geometry Measures of Nonlinearity for Filtering with Nonlinear Dynamic and Linear Measurement Mod-

els". *Proc. Signal and Data Processing of Small Targets*, volume 6699. San Diego, CA, USA., August 28-30 2007.

27. Bain, Dan, Alan Crisan. *Fundamentals of Stochastic Filtering*. Stochastic Modeling and Applied Probability. Springer, 60 edition, 2009.

28. Bashi, AS, VP Jilkov, XR Li, and H Chen. "Distributed implementations of particle filters". *Proc. Sixth International Conference*, 1164–1171, 2003. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.61.9223\&amp;rep=rep1\&amp;type=pdf`.

29. Basseville, M. "Distance Measures for Signal Processing and Pattern Recognition vol. 18, pp. 349Ű369, 1989." *IEEE Transactions on Signal Processing*, 18:349 – 369, 1989.

30. Bedworth, M. D., and J. O'Brien. "The omnibus model: A new architecture for data fusion? In Proceedings of the". *2nd International Conference on Information Fusion*. 1999.

31. Ben Upcroft, Tobias Kaupp Matthew Ridley Lee-Ling Ong Suresh Kumar Tim Bailey Fabio Ramos Salah Sukkarieh, Bertrand Douillard and Hugh Durrant-Whyte. "Non-Gaussian State Estimation in an Outdoor Decentralised Sensor Network". *45th IEEE Conference on Decision and Control*, 366 – 372. San Diego, CA, 13-15 Dec. 2006.

32. Benjamin Noack, Marcus Baum, and Uwe D. Hanebeck. "Covariance Intersection in Nonlinear Estimation Based on Pseudo Gaussian Densities". *Proceedings of the 14th International Conference on Information Fusion (FUSION)*, 1–8. Chicago, IL, 5-8 July 2011.

33. Bergman, N. *Recursive Bayesian Estimation: Navigation and Tracking Applications*. Ph.d. dissertation, Linkoping University, Department of Electrical Engineering, 1999.

34. Berlinet, A. and I. Vajda. "About the Asymptotic Accuracy of Barron Density Estimates". *IEEE Transactions on Information Theory,*, 44(3):999 – 1009, 1999.

35. Bhattacharya, A. "On A Measure of Divergence Between Two Statistical Populations Defined by Their Probability Distribution". *Bulletin of the Calcutta Mathematical Society*, 35:99 – 110, 1943.

36. Bickel, P. J. and M. Roseblatt. "On Some Global Measures of the Deviations of Density Function Estimates". *Annals of Statistics*, 1:1071Ű1095, 1973.

37. Bingham, N. H. "Positive Definite Functions on Spheres". *Proc. Camb. Phil. Soc.*, 73:145–156, 1973.

38. Bishop, Christopher M. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2 edition, 2006.

39. Blackman, S.S. *Multiple Target Tracking with Radar Applications*. Artech House Inc., 1986.

40. Blahut, Richard E. *Principles and Practice of Information Theory*. Addison-Wesley, 1991.

41. Bochardt, Ottmar, Ryan Calhoun, Jeffrey Uhlmann, and Simon Julier. "Generalized Information Representation and Compression Using Covariance Union". *2006 9th International Conference on Information Fusion*, 1–7, July 2006. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4086059`.

42. Boothby, William M. *An Introduction to Differentiable Manifolds and Riemannian Geometry*. Academic Press, 2002.

43. Borkar, M., V. Cevher, and J.H. McClellan. "Estimating Target State Distributions In a Distributed Sensor Network Using a Monte-Carlo Approach". *2005 IEEE Workshop on Machine Learning for Signal Processing*, 305–310, 2005. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1532919`.

44. Bosch, P.P.J., Papp, Z., Sijs, J., Lazar, M. "Proceedings of the 17th IEEE International Conference on Control Applications". *An overview of noncentralized Kalman filters*. 2008.

45. Boyd, J. R. . "A discourse on winning and losing", 1987. Presentation slides.

46. Branko Ristic, Neil Gordan, Sanjeev Arulampalam. *Beyond the Kalman Filter - Particle Filters for Tracking Applications*. Artech House Publishers, 2004.

47. Bravo, F.G., Alberto Vale, and M.I. Ribeiro. "Particle-filter approach and motion strategy for cooperative localization". Citeseer, 2006. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.96.3593\&amp;rep=rep1\&amp;type=pdf`.

48. Brigo, D. "On SDE with marginal laws evolving in finite dimensional exponential families". *Statist. Prob. Lett*, 49:127–134, 2000.

49. Brigo, Damiano, Contributors Damiano Brigo, Frédéric Cérou, Fabien Campillo, Marc Joannides, François LeGland, and Laurent Mevel. "Nonlinear Filtering: Stochastic Analysis and Numerical Methods". Final Report: ARO Grant DAAH04, October 1996.

50. Brigo, Damiano, Francois Le Gland, and Bernard Hanzon. "The Exponential Projection Filter and the Selection of the Exponential Family". *Proceedings of the 2nd Portuguese Conference on Automatic Control*, 251–256. 1996.

51. Brigo, Damiano and Francois LeGland. "A finite dimensional filter with exponential conditional density". *Proceedings of the 36th IEEE Conference on Decision and Control*, 1643–1644. San Diego, CA USA, December 10-12 1997.

52. Brown, Robert G. and Patrick Y.C. Hwang. *Introduction To Random Signals and Applied Kalman Filtering*. John Wiley & Sons, 1997.

53. Bryson, Mitch and Salah Sukkarieh. "Bearing-Only SLAM for an Airborne Vehicle". *Australian Conference on Robotics and Automation*. 2005.

54. Buss, S. R., and Fillmore, J. P. "Spherical averages and applications to spherical splines and interpolation". *ACM Transactions on Graphics*, 20(2):95–126, 2001.

55. Cappe, Olivier, Simon J. Godsill, and Eric Moulines. "An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo". *Proceedings of the IEEE*, 95(5):899–924, May 2007. ISSN 0018-9219. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4266870`.

56. Carmi, F. Godsill S.J., A. Septier. "The Gaussian mixture MCMC particle algorithm for dynamic cluster tracking". *12th International Conference on Information Fusion*, 1179 – 1186. Seattle, WA, 6-9 July 2009.

57. do Carmo, M.P. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, N.J., 1976.

58. Catlin, Donald E. *Estimation, Control, and the Discrete Kalman Filter*. Springer-Verlag, 1989.

59. Cesar Bolzani de Campos Ferreira, Julio and Jacques Waldmann. "Covariance intersection-based sensor fusion for sounding rocket tracking and impact area prediction". *Control Engineering Practice*, 15(4):389–409, April 2007. ISSN 09670661. URL `http://linkinghub.elsevier.com/retrieve/pii/S0967066106001250`.

60. Cha, Sung-Hyuk. "Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions 1:4 (2007)." *International Journal of Mathematical Models and Methods in Applied Sciences*, 1:300 – 307, 2007.

61. Charles F. Stevens. *The Six Core Theories of Modern Physics*. MIT, 1995.

62. Che-Bin Liu, Ruei-Sung Lin, Ming-Hsuan Yang, Narendra Ahuja, Stephen Levinson. "Object Tracking Using Globally Coordinated Nonlinear Manifolds". *18th International Conference on Pattern Recognition (ICPR)*, volume 1. 2006.

63. Chen, Zhe. *Bayesian Filtering: From Kalman Filters to Particle Filters and Beyond*. Technical report, Unknown, 2003.

64. Chong, Chee-yee and Shozo Mori. "Graphical Models for Nonlinear Distributed Estimation". *International Conference on Information Fusion*. Stockholm, Sweden, 2004. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.64.2177`.

65. Coates, Mark. “Distributed particle filters for sensor networks”. *on Information processing in sensor networks*, 26:1–16, 2004. URL `http://portal.acm.org/citation.cfm?id=984622.984637`.

66. Comaniciu, Ramesh V. Meer P., D. “Kernel-Based Object Tracking”. *IEEE Transactions on Pattern Analysis and Machine Learning*, 25(5):564 – 577, 2003.

67. Corcuera, J. M., and Kendall, W. S. “Riemannian Barycentres and Geodesic Convexity”. *Mathematical proceedings of the Cambridge Philosophical Society*, 127(2):253–269, 1999.

68. Cover, T. and J. A. Thomas. *Elements of Information Theory*. Wiley, 1991.

69. Crossley, Martin D. *Essential Topology*. Springer, 2005.

70. Csiszár, I. “Generalized Cutoff Rates and Rényi Information Measures”. *IEEE Trans. Inform. Theory*, 41(1):26–34, Jan. 1995.

71. D. Brigo, B. Hanzon and F. LeGland. “A differential geometric approach to nonlinear filtering: the projection filter”. *IEEE Trans. Automatic Control*, 43(2):247–252, 1998.

72. D. Clark, S. Julier, R. Mahler, and B. Ristic. “Robust multi-object sensor fusion with unknown correlations”. *Proceedings of the SSPD*. September 2010.

73. D. Dubois and H. Prade. *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, 1980.

74. Damiano Brigo, Bernard Hanzon and Francois LeGland. *A differential geometric approach to nonlinear filtering: the projection filter*. Tech Report RR-2598, INRIA, June 1995.

75. Damiano Brigo, Bernard Hanzon and Francois LeGland. *On the relationship between assumed density filters and projection filters*. Tech Report TI 7-96-18, Tinbergen Institute, Amsterdam, February 1996.

76. Damiano Brigo, Bernard Hanzon and Francois LeGland. “Approximate filtering by projection on the manifold of exponential densities”. *Bernoulli*, 5:495–543, 1999.

77. Damiano Brigo, Giovanni Pistone. “Projecting the Fokker-Planck Equation onto a finite dimensional exponential family”. *arXiv:0901.1308v1*, 1–20, January 2009.

78. Dasarathy, B. “Sensor Fusion Potential Exploitation Ű Innovative Architectures and Illustrative Applications”. *IEEE Proceedings*, volume 85, 24–38. 1997.

79. David Malako. “Bayes Offers a ’New Way’ to Make Sense of Numbers”. *Comments in: Science*, 286:1460–1466, November 1999.

80. Deza, Elena and Michel Marie Deza. *Dictionary of Distances*. Elsevier, 2006.

81. Dirk Tenne and Tarunraj Singh. “Circular Filters for Target Tracking in 3D”. *Citeseer*, 1:370–376, 2004. URL `http://citeseerx.ist.psu.edu/`

viewdoc/download?doi=10.1.1.1.1662\&amp;rep=rep1\&amp;type=pdf.

82. Do Carmo, Manfredo. *Riemannian Geometry*. Mathematics: Theory and Applications. Birkhäuser Boston, 2nd edition, 1988.

83. Dorje C. Brody and Lane P. Hughston. "Interest Rates and Information Geometry". *The Royal Society of London*, 457:1343 – 1363, 2001.

84. Doucet, A., Godsill, S., Andrieu, C. "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering". *Statistical Computing*, 10(3):197–208, 2000.

85. Dr. Oxley, M. "Course notes from MATH-831 Optimization by Vector Space Methods". unpublished, June - August 2008.

86. Dr. Wood, A. "Course Notes from MATH-600 Mathematical Analysis". Unpublished, Oct. - Dec. 2007.

87. Dr. Wood, A. "Course Notes from MATH-705 Linear Functional Analysis", March thru June 2008.

88. Duda, Hart P.E. Stork D.G., R.O. *Pattern Classification*. Wiley - Interscience, 2001.

89. Durrant-Whyte, H and Mike E Stevens. "Data fusion in decentralised sensing networks". 1–9. Montreal, Canada, 2001. URL http://ieee-aess.org/isif/sites/default/files/proceedings/fusion01CD/fusion/searchengine/pdf/WeA33.pdf.

90. Durrant-Whyte, H.F., B.Y.S. Rao, and H. Hu. "Toward a fully decentralized architecture for multi-sensor data fusion". *Proceedings., IEEE International Conference on Robotics and Automation*, 1331–1336, 1991. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=126185.

91. Durrant-Whyte, Hugh. "Data Fusion in Sensor Networks". *Advanced Video and Signal Based Surveillance, IEEE Conference on*, 0:39, 2006.

92. E. Taropa, V.P. Srini, L. Wong-Jong, H. Tack-Don. "Data Fusion Applied to Autonomous Ground Vehicles". *The 8th International Conference on Advanced Communication Technology, (ICACT)*, 744–749, May 2006.

93. Eduardo Montijano, Sonia Martinez, and Carlos Saques. "Distributed Robust Data Fusion Based on Dynamic Voting". *IEEE International Conference on Robotics and Automation*, 5893–5898. Shanghai International Conference Center, 9-13 May 2011.

94. Efatmaneshnik, M., A. Tabatabaei Balaei, N. Alam, and A. Dempster. "A Modified Multidimensional Scaling with Embedded Particle Filter Algorithm for Cooperative Positioning of Vehicular Networks". *in IEEE International Conference on Vehicular Electronics and Safety*. India, 2009.

95. Efron, B. "Defining the curvature of a statistical problem (with applications to second order efficiency)". *Ann. Stats.*, 6:1189–1242, 1975.

96. Émery, Michel. *Stochastic Calculus in Manifolds*. Springer-Verlag, Berlin, 1989. With an appendix by P. -A. Meyer.

97. Engelking, R. *General Topology*, volume 6 of *Sigma Series in Pure Mathematics*. Heldermann, Berlin, revised and completed edition, 1989.

98. F. Daum. "Nonlinear Filters: Beyond the Kalman Filter". *IEEE Aerospace and Electronics System Magazine*, 28:57–69, August 2005.

99. Farrell, Chimbar, William J III Ganesh. "Generalized Chernoff Fusion Approximation for Practical Distributed Data Fusion". *12th International Conference on Information Fusion*, 555–562. Seattle, WA, USA, 2009.

100. Felix Opitz. "The Differential Geometric View of Statistics and Estimation". *4th German Workshop on Sensor Data Fusion*. ISIF, 2010.

101. Florentin Smarandache & Jean Dezert. *Advances and Applications of DSmT for Information Fusion - Collected Works*. ProQuest Information & Learning, 2006.

102. Fox, Dieter. "KLD-Sampling: Adaptive Particle Filters". *In Advances in Neural Information Processing Systems*, 713–720. MIT Press.Cambridge, MA, 2001.

103. Franken, D. and A. Hupper. "Improved Fast Covariance Intersection for Distributed Data Fusion". *2005 7th International Conference on Information Fusion*, 154–160. Ieee, Seoul, Korea, 2005. ISBN 0-7803-9286-8. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1591849`.

104. Freeden, W., Gervens, T. and Schreiner, M. *Constructive Approximation on the Sphere*. Oxford U, 1998.

105. G. Kantor and S. Singh. "Preliminary results in range-only localization and mapping". *In Proc. of IEEE Conf. on Robotics and Automation*. 2002.

106. Gardner, S B and J K Uhlmann. "A Decentralized Data Fusion Framework for Horizontal Integration of Intelligence Data". *International Conference on Intelligence Analysis*, Ci, 1–4. McLean, Virginia, 2005. URL `https://analysis.mitre.org/proceedings/Final\_Papers\_Files/69\_Camera\_Ready\_Paper.pdf`.

107. Garrick, Ing. *Distributed Particle Filters for Object Tracking in Sensor Networks*. Ph.d dissertation, McGill University, 2005.

108. Gordan, D. Salmond, N. and A.F.M. "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation". *Radar and Signal Processing, IEE Proceedings F*, 140:107 – 113, 1993.

109. Grewal, Mohinder S. and Angus P. Andrews. *Kalman Filtering - Theory and Practice Using MATLAB*. John Wiley & Sons, 2001.

110. Grime, S. "Data fusion in decentralized sensor networks". *Control Engineering Practice*, 2(5):849–863, October 1994.

111. Grime, S. and H. Durrant-Whyte. “Communication in decentralized systems”. *IFAC, Control Engineering Practice*, 2:849 – 863, 1994.

112. Grime, S and H Durrant-whyte. “Data fusion in decentralized sensor networks”. *Control Engineering Practice*, 2(5):849–863, 1994.

113. H. Durrant-Whyte, M. Stevens and E. Nettleton. “Data Fusion in Decentralised Sensing Networks”. *In Proceedings of Fusion*, 302Ű307. July 2001.

114. H. Karcher. “Riemannian Center of Mass and Mollifier Smoothing”. *Communications in Pure and Applied Mathematics*, 30(5):509–541, 1977.

115. H. Q. Liu, H. C. So, F. K. W. Chan, and K. W. K. Lui. “Distributed Particle Filter for Target Tracking in Sensor Networks”. *Progress in Electromagnetics Research*, 11:171–182, 2009.

116. Hall, David L. *Mathematical Techniques in Multisensor Data Fusion*. Artech House Publishers, 1992.

117. Hall, D.L. and J. Linas. *Handbook of Multisensor Data Fusion*. CRC Press, May 2001.

118. Handshin, J.E. and D.Q. Mayne. “Monte Carlo Techniques to Estimate the Conditional Expectation in Multi-Stage Non-Linear Filtering”. *International Journal of Control*, 9:547 – 559, 1969.

119. Hoffman, Kenneth and Ray Kunze. *Linear Algebra*. Prentice Hall, 1971.

120. Hoffmann, Gabriel M., Steven L. Wasl, and Claire J. Tomlin. “Mutual information methods with particle filters for mobile sensor network control”. *in IEEE Conference on Decision and Control*, 1019–1024. 2006.

121. Hurley, M B. “An Information Theoretic Justification for Covariance Intersection and Its Generalization”. *Proc. of the 5th International Conference on Information Fusion 2002, Annapolis, Maryland, USA*, 1:505–511, 2002.

122. Hurley, M.B. *An Information-Theoretic Justification for Covariance Intersection and Its Generalizations*. Technical Report ESC-TR-2000-071, Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, MA, 1 August 2001.

123. Ihler, Alexander, John Fisher, John W. Fisher III, Alan Willsky, and Randolph Moses. “Nonparametric Belief Propagation for Self-Calibration in Sensor Networks”. *In Proceedings of the Third International Symposium on Information Processing in Sensor Networks*, 225–233. 2004. URL `http://ttic.uchicago.edu/~ihler/papers/ipsn04.pdf`.

124. Iltis, R. A. “State estimation using an approximate reduced statistics algorith”. *IEEE Transactions on Aerospace and Electronic Systems*, 35(4):1161Ű1172, October 1999.

125. Ito K., Xiong K. "Gaussian Filters for Nonlinear Filtering Problems". *Automatica*, 45:910 – 927, 2000.

126. J. Djugash and S. Singh,. "A robust method of localization and mapping using only range". International Symposium on Experimental Robotics, July 2008.

127. Jaward, M.H., D. Bull, and N. Canagarajah. "Distributed Tracking with Sequential Monte Carlo Methods for Manoeuvrable Sensors". *2006 IEEE Nonlinear Statistical Signal Processing Workshop*, 113–116, September 2006. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4378832`.

128. Jazwinski, Andrew H. *Stochastic Processes and Filtering Theory*. Dover, 1970.

129. J.E. Handshin. "Monte Carlo techniques for prediction and filtering of nonlinear stochastic processes". *Automatica*, 6:555–563, 1970.

130. Jeffreys, H. "An invariant form for the prior probability in extimation problems". *Proceedings of the Royal Statistical Society*, volume 186 of *Series A*, 453–461. London, 1946.

131. Jitendra R. Raol. *Multi-Sensor Data Fusion with MATLAB*. Engineering. CRC Press, 6000 Broken Sound Parkway NW, Suite 300, 2010.

132. J.K. Uhlmann. *Dynamic Map Building and Localization for Autonomous Vehicles*. Ph.D. thesis, University of Oxford, 1995.

133. J.M. Hammersley and K.W. Morton. "Poor manŠs Monte Carlo". *Journal of the Royal Statistical Society*, 16–23, 1954.

134. Johnson, O. and C. Vignat. "Some results concerning maximum Rényi entropy". *Annales de l'Institut Henri Poincarï Probability and Statistics*, 43(3):339–351, 2007.

135. Johnson, Don and Sinanovic, Sinan. "Symmetrization of the Kullback-Leibler Distance". *IEEE Transactions on Information Theory*, March 2001.

136. Julier, Simon J. "Estimating and Exploiting the Degree of Independent Information in Distributed Data Fusion". *12th International Conference on Information Fusion*, 1–8. Seattle, WA, USA, July 2009. URL `http://eprints.ucl.ac.uk/15814/1/15814.pdf`.

137. Julier, Simon J. and Jeffrey K. Uhlmann. "Unscented Filtering and Nonlinear Estimation". *Proceedings of the IEEE*, 92(3):401–422, March 2004.

138. Julier, S.J. "An Empirical Study Into the Use of Chernoff Information for Robust, Distributed Fusion of Gaussian Mixture Models". *9th International Conference on Information Fusion*. 2006.

139. Julier, SJ, Tim Bailey, and JK Uhlmann. "Using Exponential Mixture Models for Suboptimal Distributed Data Fusion". *2006 IEEE Nonlinear Statistical Signal*, (1),

2006. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.125.9269\&amp;rep=rep1\&amp;type=pdf`.

140. Julier S.J., H.F. Durrant-Whyte, J.K. Uhlmann. "A New Method for The Nonlinear Transformation of Means and Covariances in Filters and Estimators". *IEEE Transactions on Automatic Control*, 45:477 – 482, 2000.

141. Julier S.J., J.K. Uhlmann. *A General Method for Approximating Nonlinear Transformations of Probability Distributions*. Technical report, University of Oxford, 1996.

142. Julier S.J., J.K. Uhlmann. "Reduced Sigma-Point Filters for the Propogation of Means and Covariances Through Nonlinear Transformations". *Proceedings of the IEEE American Control Conference*, 887–892. 2002.

143. Julier S.J., J.K. Uhlmann. "The Scaled Unscented Transformation". *Proceedings of the IEEE American Control Conference*, 4555 – 4559. 2002.

144. Junghyun Kwon, F. C. Park, Minseok Choi and Changmook Chun. "Particle filtering on the Euclidean group: framework and applications". *Robotica*, 25:725–737, June 2007.

145. K.A. Krakowski, K. Hüper and J.H. Manton. "On the Computation of the Karcher Mean on Spheres and Special Orthogonal Groups". *RoboMat, Workshop on Robotics and Mathematics*. Coimbra, Portugal, 17Ű19 September 2007.

146. Kailath, T. "The divergence and Bhattacharrya distance in signal selection." *IEEE Trans. Comm.*, 15:52–60, 1967.

147. Kailath, T. *Lectures on Wiener and Kalman Filtering*. Springer-Verlag, New York, 1981.

148. Kalman, R.E. "A New Approach to Linear Filterining and Prediction Problems". *Transactions on AMSE, Journal of Basic Engineering*, 82:34 – 45, March 1960.

149. Kass, R. and P. Vos. *Geometrical Foundations of Asymptotic Inference*. Wiley, New York, 1997.

150. KC Chang, Chee-Yee Chong, and Shozo Mori. "Analytical and Computational Evaluation of Scalable Distributed Fusion Algorithms". *IEEE transactions on Aerospace and Electronic Systems*, 46(4):2022–2034, October 2010.

151. Kevin Patrick Murphy. "Mixture Models". Online, November 2006. URL `http://www.cs.ubc.ca/~murphyk/`. A collection of notes available online.

152. Kim, Jonghyuk and Salah Sukkarieh. "Autonomous Airborne Navigation in Unknown Terrain Environments". *IEEE Transactions on Aerospace Electronic Systems*, 40(3):1031 – 1045, July 2003.

153. Kitagawa, G. "Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models". *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.

154. Kobayashi, S. and Nomizu, K. *Foundations of Differential Geometry, vol. 2.* Interscience tracts in pure and applied mathematics. Interscience Publishers, 1969.

155. Kulhavy, R. “Recursive nonlinear estimation: A geometric approach”. *Automatica*, 26:545–555, 1990.

156. Kulhavy, R. “Recursive nonlinear estimation: Geometry of a space of posterior densities”. *Automatica*, 28(2):313Ű323, 1992.

157. Kulhavy, R. “A geometric approach to statistical estimation”. *Proceedings of the 34th IEEE Conference on Decision and Control*, volume 2, 1097–1102. December 1995.

158. Kulhavy, Rudolf. “Can we preserve the structure of recursive Bayesian estimation in a limited-dimensional implementation?” *12th World Congress IFAC*, 225–228. IFAC, Sydney Australia, 1993.

159. Kulhavy, Rudolph. “Recursive Bayesian Estimation Under Memory Limitations”. *Kybernetika*, 26:1–16, 1990.

160. Kwon, Junghyun and F.C. Park. “A Geometric Approach to Particle Filtering-Based Visual Tracking”. *cv.snu.ac.kr*, 1998. URL `http://cv.snu.ac.kr/jhkwon/data\%5Curai08.pdf`.

161. Kwon, Junghyun and Frank Chongwoo Park. “Visual Tracking via Particle Filtering on the Affine Group”. *The International Journal of Robotics Research*, 29(198):1–6, August 2008.

162. Kyle Gallivan, Xiuwen Liu, Anuj Srivastava. “Efficient Algorithms For Inferences On Grassmann Manifolds”. *Proceedings of 12 th IEEE Workshop on Statistical Signal Processing*. 2003.

163. Langelaan, Jack and Steve Rock. “Navigation of Small UAVs Operating in Forests”. *Proc AIAA Guidance, Navigation, and Control Conference*. August 16 - 19 2004.

164. Lee, Chan-Su and Ahmed Elgammal. “Modeling View and Posture Manifolds for Tracking”. *2007 IEEE 11th International Conference on Computer Vision*, 1–8, October 2007. ISSN 1550-5499. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4409030`.

165. Lee, Deok-Jin. “Unscented Information Filtering for Distributed Estimation and Multiple Sensor Fusion”. *AIAA Guidance, Navigation and Control Conference*. 2008.

166. Lee, John M. *Riemannian Manifolds - An Introduction to Curvature*. Springer, 1997.

167. Lee-Ling Ong. *Non-Gaussian Representations for Decentralised Bayesian Estimation*. Ph.d dissertation, The University of Sydney, March 2007.

168. Lee-Ling Ong, Ben Upcroft, Tim Bailey and Hugh Durrant-Whyte. "Decentralised Particle Filtering for Multiple Target Tracking in Wireless Sensor Networks". *Fusion 2008*, 1–8, 2008.

169. Lee-Ling Ong, Ben Upcroft Suresh Kumar-Tim Bailey Salah Sukkarieh, Matthew Ridley and Hugh Durrant-Whyte. "A Comparison of Probabilistic Representations for Decentralised Data Fusion". *ISSNIP*. 2005.

170. Lee-Ling Ong, Ben Upcroft, Tim Bailey, Matthew Ridley, Salah Sukkarieh and Hugh Durrant-Whyte. "A Decentralised Particle Filtering Algorithm for Multi-Target Tracking Across Multiple Flight Vehicles". *International Conference on Intelligent Robots and Systems*. 2006.

171. LeGland, Francois. "Stability and Approximation of Nonlinear Filters: An Information Theoretic Approach". *Proceedings of the 38th IEEE Conference on Decision and Control*, 1889–1894. 7-10 December 1999.

172. Liggins, Hall D.L. Llinas J., M.E. *Handbook of Multisensor Data Fusion - Theory and Practice*. CRC Press LLC, 2nd edition, 2009.

173. Liggins, M.E., I. Kadar, M.G. Alford, V. Vannicola, and S. Thomopoulos. "Distributed Fusion Architectures and Algorithms for Target Tracking". *Proceedings of the IEEE*, 85(1):95–107, January 1997. ISSN 0018-9219. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=554211`.

174. Lin, J. "Divergence Measures Based on the Shannon Entropy". *IEEE Trans. Infor. Theory*, 37:145–151, 1991.

175. Liu, Jun S. and Rong Chen. "Sequential Monte Carlo Methods for Dynamic Systems". *Journal of the American Statistical Association*, 93:1032 – 1044, 1998.

176. Lopes, C. G., Cattivelli, F. S., Sayed, A. H. "Diffusion strategies for distributed Kalman filtering: formulation and performance analysis." *Cognitive Information Processing*, 2008.

177. M. Fréchet. "Sur l'extension de certaines évaluations statistiques de petits echantillons". *Rev. Inst. Int. Statist*, 11:128–205, 1943.

178. M. Fréchet. "L'intégrale abstraite d'une fonction abstraite d'une variable abstrait et son application á la moyenne d'un élément aléatorie de nature quelconque". *Revue Scientifique*, 483–512, 1944.

179. M. Fréchet. "Les éléments aléatories de nature qualconque dans un espace distancié". *Ann. Inst. Henri Poincaré*, 10:215–310, 1948.

180. Ma, Lian Li Yan Liu Limin Sun Jian. "Lightweight Particle Filters based Localization Algorithm for Mobile Sensor Networks". *2008 Second International Conference on Sensor Technologies and Applications*, 135–140. 25-31, August 2008.

181. Mahler, Ronald. “Optimal/robust distributed data fusion: a unified approach”. *Signal Processing, Sensor Fusion, and Target Recognition IX*, volume 4052, 128–138. SPIE, Orlando, FL, USA, 2000.

182. Makarenko, a and H Durrantwhyte. “Decentralized Bayesian Algorithms for Active Sensor Networks”. *Information Fusion*, 7(4):418–433, December 2006. ISSN 15662535. URL `http://linkinghub.elsevier.com/retrieve/pii/S1566253505000813`.

183. Mallick, M. and B. F. La Scala. “Differential Geometry Measures of Nonlinearity for Ground Moving target Indicator (GMTI) Filtering”. *Proc. 2005 International Conference on Information Fusion*. Philadelphia, USA, July 25-29, 2005.

184. Mallick, M. and B. F. La Scala. “Differential Geometry Measures of Nonlinearity for the Video Tracking Problem”. *Proc. Signal Processing, Sensor Fusion, and Target Recognition, XV*. Orlando, FL, USA, 17-21 April 2006.

185. Mallick, M. and B. F. La Scala. “Differential geometry measures of nonlinearity for the video filtering problem”. *Proc. Signal and Data Processing of Small Targets*, volume 6969. Orlando, FL, USA, March 18-20 2008.

186. Manyika, J. and H.F. Durrant-Whyte. *Data Fusion and Sensor Management*. Prentice Hall, 1994.

187. Marc Reinhardt, Benjamin Noack, Marcus Baum, and Uwe D. Hanebeck. “Analysis of Set-theoretic and Stochastic Models for Fusion under Unknown Correlations”. *IEEE 14th International Conference on Information Fusion (FUSION)*, 1–8. Chicago, IL, 5-8 July 2011.

188. Maria-Joao Rendas and Jose Manuel Leitao. “Rumor-Robust Distributed Data Fusion”. *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 230–235. University of Utah, Salt Lake City, UT, USA, 5-7 September 2010.

189. Mark A. Davenport, Chinmay Hegde, Marco F. Duarte, Richard G. Baraniuk. “Joint Manifolds for Data Fusion”. *IEEE Transactions on Image Processing*, 19(10):2580–2594, October 2010.

190. Markin, M., C. Harris, M. Bernhardt, J. Austin, M. Bedworth, P. Greenway,R. Johnston, et al. “Technology foresight on data fusion and data processing.” *The Royal Aeronautical Society*, 1997.

191. Martin, T.W. and K.C. Chang. “A Distributed Data Fusion Approach for Mobile Ad Hoc Networks”. *2005 7th International Conference on Information Fusion*, 1062–1069, 2005. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1591975`.

192. Maybeck, Peter S. *Stochastic Models Estimation and Control I*. Academic Press, Inc., Orlando, Florida 32887, 1979.

193. Maybeck, Peter S. *Stochastic Models Estimation and Control, II*. Academic Press, Inc., Orlando, Florida 32887, 1982.

194. van der Merwe, Rudolph and Eric Wang. "Sigma-Point Kalman Filters for Nonlinear Estimation and Sensor-Fusion with Applications to Integrated Navigation". *Proc. of AIAA Guidance, Navigation, and Control Conference*. August 16 - 19 2004.

195. Metropolis, N. and S. Ulam. "The Monte Carlo Method". *Journal of American Statistical Association*, 44(247):335–341, September 1949.

196. Miroslav Simandl. "Lecture Notes on State Estimation of Nonlinear Non-Gaussian Stochastic Systems". online, 2006.

197. Mitchell, H.B. *Multi-Sensor Data Fusion - An Introduction*. Springer, 2007.

198. M.N. Rosenbluth and A.W. Rosenbluth. "Monte Carlo calculation of the average extension of molecular chains". *Journal of Chemical Physics*, 23, 1956.

199. Moakher, Maher. "On the Averaging of Symmetric Positive-Definite Tensors". *Journal of Elasticity*, 82:273–296, 2006.

200. Mohammad Sadegh and Mohebbi Nazar. "A Comparative Study of Different Kalman Filtering Methods in Multi-Sensor Data Fusion". *Proceedings of the International MultiConference of Engineers and Computer Scientists*. 2009.

201. Mohinder S. Grewal, Lawrence R. Weill, and Angus P. Andrews. *Global Positioning Systems, Inertial Navigation, and Integration*. John Wiley and Sons, Inc., Hoboken, New Jersey, 2nd edition, 2007.

202. Munkres, James R. *Topology*. Prentice Hall, 2nd edition, 2000.

203. Mutambara, Arthur G.O. *Decentralized Estimation and Control for Multisensor Systems*. CRC Press LLC, 1998.

204. Naylor, Sell G.R., A.W. *Linear Operator Theory in Engineering and Science*. Springer, 1982.

205. Nettleton, E, H Durrant-Whyte, and S Sukkarieh. "A robust architecture for decentralised data fusion". *Proc. of the International Conference on*, 2003. URL `http://scholar.google.com/scholar?hl=en\&btnG=Search\&q=intitle:A+Robust+Architecture+for+Decentralised+Data+Fusion\#0`.

206. Nettleton, E W and H F Durrant-whyte. "Delayed and asequent data in decentralised sensing networks". *in Sensor Fusion and Decentralized Control in Robotic Systems*, 4571(IV):1–9, 2001.

207. Noakes, L. "Sample Statistics and Curved Spaces". Unpublished.

208. Ojha, Unnati, Mo-Yuen Chow, Timothy Chang, and Janice Daniel. "Analysis on the Kalman Filter Performance in GPS/INS integration at Different Noise Levels,

Sampling Periods, and Curvatures". *2009 35th Annual Conference of IEEE Industrial Electronics*, 2975–2980, 2009. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5415281`.

209. Olfati-Saber, R. "Distributed kalman filtering for sensor networks". *Proceedings of the 46th IEEE Conference on Decision and Control*, 5492. New Orleans, LA, 12-14 December 2007 2007.

210. Oller, J. M., AND Corcuera, J. M. "Intrinsic analysis of statistical estimation." *The Annals of Statistics*, 23(5):1562Ű1581, 1995.

211. Ong, Lee-ling, Ben Upcroft, Matthew Ridley, Tim Bailey, Salah Sukkarieh, and Hugh Durrant-Whyte. "Consistent methods for Decentralised Data Fusion using Particle Filters". *2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, (d):85–91, September 2006. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4042021`.

212. Ong, Upcroft B. Ridley M.F. Bailey T.A. Sukkarieh-S. & Durrant-Whyte H.F., L. "Decentralised Data Fusion with Particles". *Proceedings Australiasian Conference on Robotics and Automation (ACRA)*. 2005.

213. Opitz, Felix. "From Differential to Information Geometry". *IEEE 2nd International Workshop on Cognitive Information Processing*. 2010.

214. Ottmar Bochardt and Jeffrey Uhlmann. "On the Equivalence of the General Covariance Union (GCU) and Minimum Enclosing Ellipsoid (MEE) Problems". *CoRR*, abs/1012.4795:1–14, 2010.

215. Oxley, M. E. "Course Notes from MATH 621 Linear Algebra", Fall Quarter 2007.

216. Pachter, Meir and Alec Porter. "Bearings-only Measurements for INS Aiding: The Three-Dimensional Case". *Proceedings of the 2003 AIAA Guidance, Navigation and Control Conference*. 2003. AIAA paper number 2003-5354.

217. Pao, Lucy Y. "Distributed Multisensor Fusion". *AIAA Journal of Guidance Navigation and Control*, 1994.

218. Papoulis, Athanasios and S. Unnikrishna Pillai. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, New York, NY 10020, 2002.

219. Pardo, Leandro. *Statistical Inference Based on Divergence Measures*, volume 185 of *Statistics: Textbooks and Monographs*. Chapman and Hall/CRC, 2006.

220. Park, F.C. "Visual tracking via geometric particle filtering on the affine group with optimal importance functions". *IEEE Conference on Computer Vision and Pattern Recognition*, 991–998, June 2009. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5206501`.

221. Park, Frank C. "Visual tracking via particle filtering on the affine group". *2008 International Conference on Information and Automation*, 997–1002, June 2008. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4608144`.

222. Parzen, E. "On Estimation of a Probability Density and Mode". *Ann. Mathematical Statistics and Computing*, 33(3):1065–1076, 1962.

223. Paul Bui Quang, Christian Musso, and Francois Le Gland. "An Insight into the Issue of Dimensionality in Particle Filtering". *IEEE 13th Conference on Information Fusion (FUSION)*, 1–8. Edinburgh, Scotland, 26-29 July 2010.

224. Pavan Turaga, Ashok Veeraraghavan and Rama Chellappa. "Statistical Analysis on Stiefel and Grassmann manifolds with applications in computer vision". *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. 2008.

225. Pennec, Xavier. "Probabilities and Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements". *International Workshop on Nonlinear Signal and Image Processing (NSIP)*. IEEE, Antalya, Turkey, 20-23 June 1999.

226. Pennec, Xavier. "Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements". *Journal of Mathematical Imaging and Vision Research*, 25(1):127–154, July 2006.

227. Perera, L.D.L. and E. Nettleton. "Nonlinear Observability of the Centralized Multi-Vehicle SLAM Problem". *IEEE International Conference on Robotics and Automation*, May 2010.

228. Petersen, Peter. *Riemannian Geometry*. Springer, 2006.

229. P.M. Djuric, J.H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M.F. Bugallo, and J. Miguez. "Particle filtering". *IEEE Signal Processing Magazine*, 20:19–38, September 2003.

230. Pressley, Andrew. *Elementary Differential Geometry*. Springer, 2001.

231. P.S. Bullen. *Handbook of Maeans and Their Inequalities*, volume 560 of *Mathematics and Its Applications*. Kluwer Academic Publishers, 2003.

232. R. Olfati-Saber, J.A. Fax, R.M. Murray. "Consensus and Cooperation in Networked Multi-Agent Systems". *Proc. of the IEEE*, 95(1):215–233, Jan 2007.

233. Ramamurti Shankar. *Principles of Quantum Mechanics*. Plenum Press - New York and London, 1994.

234. Randal Beard, Jonathan Lawton Wynn Stirling, Jacob Gunther. "Nonlinear Projection Filter based on Galerkin Approximation". *AIAA Journal of Guidance, Control and Dynamics*, 22(2):258–266, March-April 1999.

235. Randal Douc, Eric Moulines, Olivier Cappé. “Comparison of Resampling Schemes for Particle Filtering”. *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, 64–69, September 2005.

236. Rao, C. A. “Information and the Accuracy Attainable in the Estimation of Statistical Parameters”. *Bukketin of the Calcutta Mathematical Society*, 37:81 – 91, 1945.

237. Raquet, John F. and Michael Giebner. “Navigation Using Optical Measurements of Objects at Unknown Locations”. *Proceedings of the 59th Annual Meeting of the Institute of Navigation*, 282–290. June 2003.

238. R.E. Moore. *Interval Analysis*. Prentice Hall, 1966.

239. Ren C. Luo, Ying Chih Chou, Ogst Chen. “Multisensor Fusion and Integration: Algorithms, Applications, and Future Research Directions”. *IEEE International Conference of Mechatronics and Automation*. 2007.

240. Rentmeesters, Van Dooren Gallivan, Absil and Srivastava. “An Efficient Particle Filtering Technique on the Grassmann Manifold”. *International Conference on Audio, Speech and Signal Processing (ICASSP)*. 2010.

241. Ridley, Upcroft B. Ong L. Kumar S. & Sukkarieh S, M.F. “Decentralised Data Fusion With Parzen Density Estimates”. *In Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 161 – 166. 2004.

242. Roberts, S. Reece S. “Robust, Low-Bandwidth, Multi-Vehicle Mapping”. *in Proceedings of the 2005 FUSION Conference*, 1319Ű1326. Philadelphia, PA, USA, 2005. URL `www.robots.ox.ac.uk/~sjrob/Pubs/fusion05a.pdf`.

243. Roland Siegwart and Illah R. Nourbakhsh,. *Introduction to Autonomous Mobile Robots*. The MIT Press, 2004.

244. Rosencrantz, M and Gordon, G and Thrun, S. “Decentralized Sensor Fusion with Distributed Particle Filters”. *in the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI-03*, 493–500, 2003.

245. Rudin, W. *Principles of Mathematical Analysis*. International Series in Pure and Applied Mathematic. McGraw-Hill Book Co., New York, third ed. edition, 1976.

246. Ryan, Allison and J. Karl Hedrick. “Particle Filter Based Information-Theoretic Active Sensing”. *Robotics and Autonomous Systems*, 58(5):574–584, May 2010.

247. S. Amari and A. Cichocki. “Information Geometry of Divergence Functions”. *Bulletin of the Polish Academy of Sciences Technical Sciences*, 58(1):183–195, 2010.

248. S. G. Loizou and V. Kumar. “Biologically inspired bearing-only navigation and tracking”. *Proceedings of the 46th IEEE Conference on Decision and Control*, 1386 – 1391. Dec. 2007.

249. S. J. Julier and J. K. Uhlmann. “A New Extension of the Kalman Filter to Nonlinear Systems”. *In Proc. of AeroSense: The 11th Int. Symp. on Aerospace/Defence Sensing, Simulation and Controls*, 1997.

250. S. Julier. “Fusion Without Independence”. *(keynote abstract) of the IET Seminar on Tracking and Data Fusion: Algorithms and Applications*, 1–5. 15Ű16 April 2008.

251. Sabin, J. and R. Gray. “Global Convergence and Empirical Consistency of the Generalized Lloyd Algorithm”. *IEEE Transactions on Information Theory*, 32(2):148–155, 1986.

252. Sebastian Blank, Tobias Fohst, and Karsten Berns. “A Case Study Towards Evaluation of Redundant Multi-Sensor Data Fusion”. K. Berns, C. Schindler, K. Drebler, B. Jorg, R. Kalmar, and J. Hirth (editor), *Proceedings of the 1st Commercial Vehicle Technology Symposium (CVT)*. Shaker Verlag, Kaiserslautern, Germany, March 2010.

253. Shafer, G. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

254. Sheng, Xiaohong, Y.H. Hu, and Parameswaran Ramanathan. “Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network”. *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, 181â??188. 2005. URL `http://scholar.google.com/scholar?hl=en\&btnG=Search\&q=intitle:Distributed+particle+filter+with+gmm+approximation+for+multiple+targets+localization+and+tracking+in+wireless+sensor+network\#0`.

255. Shigeyuki Morita, Katsumi Nomizu, Teruko Nagase. *Geometry of Differential Forms*. American Mathematical Society, 2001.

256. Shun-ichi Amari. “Information Geometry in Optimization, Machine Learning, and Statistical Inference”. *Frontiers in Electronics and Electrical Engineering*, 5(3):241–260, 2010.

257. Simon, Dan. *Optimal State Estimation - Kalman, Hinf, and Nonlinear Approaches*. John Wiley & Sons, 2006.

258. Simukai W. Utete. *Network Management in Decentralised Sensing Systems*. Ph.d. dissertation, University of Oxford, Division of Mathematical and Physical Sciences, January 1994.

259. Sinanovic, Sinan and Don H. Johnson. “Toward a Theory of Information Processing”. *Journal of Signal Processing*, 87:1326–1344, 2007.

260. Snoussi, H. and A. Mohammad-Djafari. “Particle Filtering on Riemannian Manifolds”. Workshop on Bayesian Inference and Maximum Entropy Methods., July 2006.

261. Snoussi, H and C Richard. "Monte Carlo Tracking on the Riemannian Manifold of Multivariate Normal Distributions". *IEEE DSP*. January 2009.

262. Soto, A. "Self Adaptive Particle Filter". *Proceedings of International Join Conference on Artificial Intelligence (IJCAI)*. 2005.

263. Speyer, J.L. "Communication and Transmission Requirements for a Decentralized Linear Quadratic-Gaussian Control Problem". *IEEE Transaction on Automatic Control*, 24(2):266–269, 1979.

264. Srivastava, Anuj and Eric Klassen. "Monte Carlo extrinsic estimators for manifold-valued parameters". *IEEE Transactions on Signal Processing*, 50(2):299–308, February 2002.

265. Srivastava, Jermyn and Joshi. "Riemannian Analysis of Probability Density Functions with Applications in Vision". *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2007.

266. Steinberg, A.N., Bowman, C.L. *Handbook of Multisensor Data Fusion*, chapter 2, Revision to the JDL data fusion model ,, 2–1 – 2–18. CRC Press, 2001.

267. Steinberg, A.N., Bowman, C.L., and White, Je. "Revisions to the JDL Data Fusion Model". *Proc. 3rd NATO/IRIS Conf.*, 1998. Quebec City, Canada.

268. Stephan Matzka and Richard Altendorfer. "A Comparison of Track-to-Track Fusion Algorithms for Automotive Sensor Fusion". *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 189–194. Seoul, Korea, August 20-22 2008. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4648063`.

269. Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, seventh edition, 2009.

270. Steven N. Thorsen and Mark E. Oxley. "A Category Theory Description of Multisensor Fusion". Belur V. Dasarathy (editor), *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, volume 5434, 261 – 269. SPIE, Bellingham, WA, 2004.

271. Stone, Lawrence D., Thomas L. Corwin, and Carl A. Barlow. *Bayesian Multiple Target Tracking*. Artech House, Inc, Norwood, MA, USA, 1999.

272. Strasdat, H, C Stachniss, M Bennewitz, and W Burgard. "Visual Bearing-Only Simultaneous Localization and Mapping with Improved Feature Matching". Kaiserslautern, Germany, 2007. URL `http://www.informatik.uni-freiburg.de/\~{}stachnis/pdf/strasdat07ams.pdf`.

273. Systems, Advanced Decision. *DISTRIBUTED MULT/TARGET MULT/SENSOR TRACKING*, chapter 8. Artech House, 1990.

274. Tenenbaum, J. B., V. De Silva, and J. C. Langford. “A global geometric framework for nonlinear dimensionality reduction”. *Science*, 2000.

275. Thomas Bengtsson, Peter Bickel, and Bo Li. “Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems”. *Collections*, 2:316–334, 2008.

276. Tianshi Chen, Thomas B. Schon, Henrik Ohlsson and Lennart Ljung. “Decentralization of Particle Filters Using Arbitrary State Decomposition”. *49th IEEE Conference on Decision and Control*, December 2010.

277. Tim Bailey, Simon Julier, and Gabriel Agamennoni. *On Conservative Fusion of Information with Unknown Non-Gaussian Dependence*. Technical report, Australian Centre for Field Robotics, University of Sydney, June 2011.

278. Tomas Svoboda, Jan Kybic, Vaclav Hlavac. *Image Processing, Analysis, and Machine Vision: A MATLAB Companion*. Cengage Learning, 2007.

279. Tompkins, Frank and Patrick J. Wolfe. “Bayesian Filtering on the Stiefel Manifold”. *2007 2nd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, 261–264, December 2007. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4498015`.

280. Tu, L.C. “Nodes Localization through Data Fusion in Sensor Network”. *19th International Conference on Advanced Information Networking and Applications (AINA’05) Volume 1 (AINA papers)*, 337–342, 2005. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1423514`.

281. Turaga, Srivastava, Veeraraghavan and Chellappa. “Statistical Analysis on Manifolds and Its Applications to Video Analysis”. *Handbook on Video Search and Mining*. Springer Verlag, 2009.

282. Tyagi, Ambrish and James W. Davis. “A Recursive Filter For Linear Systems on Riemannian Manifolds”. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8. Anchorage, Alaska, June 2008.

283. Čencov, N. N. *Statistical decision rules and optimal inference.* Nauka, Moscow, 1972. In russian, translation in "Translations of Mathematical Monographs", 53.AmericanMathematicalSociety, 1982.

284. Udo Frese. “Least Least-Square based SLAM: Linearization, Sparsity Sparsity, and Manifolds”, May 2008.

285. Uhlmann, Jeffrey K. “General data fusion for estimates with unknown cross covariances”. *Proceedings of SPIE, the International Society for Optical Engineering*, 2755:536–547, 1996.

286. Uhlmann, Jeffrey K. “Nondivergent simultaneous map building and localization using covariance intersection”. *Proceedings of SPIE*, 3087:2–11, 1997. ISSN

0277786X. URL `http://link.aip.org/link/?PSI/3087/2/1\&Agg=doi`.

287. Upcroft, B.,Kumar, S.,Ridley, M.,Bailey, T.,Sukkarieh, S.,Durrant-Whyte, H. “Rich Probabilistic Representations for Bearing Only Decentralised Data Fusion”. *7th International Conference on Information Fusion*, 1054–1061, 2005.

288. Utete, Durrant-Whyte H.F., S. “Reliability in Decentralised Data Fusion Networks”. *IEEE International Conference on Multisensor Fusion and Integration for Intelligent System*, 215 – 221, 1994.

289. Vemula, Joaquin Artes-Rodriguez Antonio, Mahesh Miguez. “A Sequential Monte Carlo Method for Target Tracking in an Asynchronous Wireless Sensor Network”. *4th Workshop on Positioning, Navigation and Communication*, 4:49–54, 2007.

290. Venkatesh, Y. V. Aand, Ko C. C. Yiyao, L. “A knowledge-based neural network for fusing edge maps of multi-sensor images”. *Information Fusion*, 2:121–133, 2001.

291. Victor Solo. “On Nonlinear State Estimation in a Riemannian Manifold”. *Joint 48th IEEE Conference on Decision and Control*, 8500–8505. Shanghai, P.R. China, December 16-18 2009.

292. Vinyals, M., J. a. Rodriguez-Aguilar, and J. Cerquides. “A Survey on Sensor Networks from a Multiagent Perspective”. *The Computer Journal*, February 2010. ISSN 0010-4620. URL `http://comjnl.oxfordjournals.org/cgi/doi/10.1093/comjnl/bxq018`.

293. Vira, S. *Multi-Agent Systems for Distributed Data Fusion in Peer-To-Peer Environment*. Master’s thesis, University of Jyvaskyla., 2002.

294. Waltz, Edward and James Llinas. *Multisensor Data Fusion*. Artech House Publishers, 1990.

295. Warner, Frank W. *Foundations of Differentiable Manifolds and Lie Groups*. Springer, 1983.

296. Wen-shiang Chen. “An Introduction to Importance Sampling Methods”, February 2002.

297. Wu, Yi, Xiaofeng Tong, Yimin Zhang, and Hanqing Lu. “BOOSTED INTERACTIVELY DISTRIBUTED PARTICLE FILTER FOR AUTOMATIC MULTI-OBJECT TRACKING”. *Science*, 2008–2011, 2008.

298. Wu, Yi, Bo Wu, Jia Liu, and Hanqing Lu. “Probabilistic tracking on Riemannian manifolds”. *2008 19th International Conference on Pattern Recognition*, 1–4, December 2008. ISSN 1051-4651. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4761046`.

299. X. Pennec. “Statistical Computing on Manifolds: From Riemannian Geometry to Computational Anatomy”. *Emerging Trends in Visual Computing*, volume 5416 of *LNCS*, 347–386. Springer, 2008.

300. Xin Tian, Yaakov Bar-Shalom, and Genshe Chen. "A No-Loss Covariance Intersection Algorithm for Track-to-Track Fusion". Oliver E. Drummond (editor), *Proc. of SPIE Signal and Data Processing of Small Targets*, volume 7698. 2010.

301. Xue, Feng, Zhong Liu, and Xiaorui Zhang. "Decentralized Information Particle Filter for Passive Tracking in Sensor Networks". *Communications and Networking in China, 2006.*, 20–22, 2006. URL `http://scholar.google.com/scholar?hl=en\&btnG=Search\&q=intitle:Decentralized+Information+Particle+Filter+for+Passive+Tracking+in+Sensor+Networks\#0`.

302. Yan Zhou and Jianxun Li. "Data Fusion of Unknown Correlations Using Internel Ellipsoidal Approximation". *The International Journal of Automatic Control*, 2856–2861, 2008.

303. Zadeh, L.A. "Fuzzy sets as a basis for a theory of possibility". *Fuzzy Sets and Systems*, 100:9–34, 1999.

304. Zhe, Chen. *Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond*. Tech report, McMaster University, June 2003. URL `http://soma.crl.mcmaster.ca/~zhechen/ieee_bayes.ps`.

305. Zheng, L. and D. N.C. Tse. "Packing spheres in the Grassman manifold: a geometric approach to the non-coherent multi-antenna channel". *IEEE Transactions on Information Theory*, 48(2):359–382, Feb. 2002.

306. Zhengdong Lu, Todd K. Leen, Rudolph van der Merwe, Sergey Frolov, and Antonio M. Baptista. *Sequential Data Assimilation with Sigma-point Kalman Filter on Low-dimensional Manifold*. Technical report, Oregon Health & Science University, Department of Computer Science & Electrical Engineering, OGI School of Science & Engineering, Portland, OR 97006, USA, June 2007.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. REPORT DATE *(DD-MM-YYYY)* 14-06-2012	2. REPORT TYPE Dissertation	**3. DATES COVERED** *(From – To)* September 2007-June 2012

4. TITLE AND SUBTITLE

Decentralized Riemannian Particle Filtering with Applications to Multi-Agent Localization

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Eilders, Martin, J., Civilian, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765 DSN: 785-3636

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/DEE/ENG/12-05

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Intentionally Left Blank

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

The primary focus of this research is to develop consistent nonlinear decentralized particle filtering approaches to the problem of multiple agent localization. A key aspect in our development is the use of Riemannian geometry to exploit the inherently non-Euclidean characteristics that are typical when considering multiple agent localization scenarios. A decentralized formulation is considered due to the practical advantages it provides over centralized fusion architectures.

Inspiration is taken from the relatively new field of information geometry and the more established research field of computer vision. Differential geometric tools such as manifolds, geodesics, tangent spaces, exponential, and logarithmic mappings are used extensively to describe probabilistic quantities. Numerous probabilistic parameterizations were identified, settling on the efficient square-root probability density function parameterization. The square-root parameterization has the benefit of allowing filter calculations to be carried out on the well studied Riemannian unit hypersphere. A key advantage for selecting the unit hypersphere is that it permits closed-form calculations, a characteristic that is not shared by current solution approaches.

Through the use of the Riemannian geometry of the unit hypersphere, we are able to demonstrate the ability to produce estimates that are not overly optimistic. Results are presented that clearly show the ability of the proposed approaches to outperform current state-of-the-art decentralized particle filtering methods. In particular, results are presented that emphasize the achievable improvement in estimation error, estimator consistency, and required computation al burden.

15. SUBJECT TERMS

Data Fusion, Riemannian Geometry, Particle Filtering, Multiagent

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Dr, John F. Raquet (ENG)
REPORT U	ABSTRACT U	c. THIS PAGE U	UU	211	**19b. TELEPHONE NUMBER** *(Include area code)* (937)255-3636x4580; email:john.raquet@afit.edu