



**TÉCNICO**  
LISBOA

# **Distributed and Centralized Observers for Sensor Fusion with Non-Gaussian Noises**

**Duarte Venâncio Leão Ribeiro da Silva**

Thesis to obtain the Master of Science Degree in

## **Electrical and Computer Engineering**

Supervisors: Prof. Daniel de Matos Silvestre  
Prof. Rita Maria Mendes de Almeida Correia da Cunha

### **Examination Committee**

Chairperson: Prof. João Manuel de Freitas Xavier  
Supervisor: Prof. Daniel de Matos Silvestre  
Member of the Committee: Prof. João Pedro Castilho Pereira Santos Gomes

**November 2024**



# Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.



# Acknowledgments

I would like to thank my parents for their unwavering friendship, encouragement, and care throughout the years. Their constant support, through both the highs and lows, has been instrumental in making this thesis possible. I would also like to extend my heartfelt thanks to my brother, and sister for their patience and encouragement along the way.

My sincere appreciation goes to my dissertation supervisor, Professor Daniel Silvestre, and my co-supervisor, Professor Rita Cunha, for their valuable guidance, knowledge, and support, which have been essential to the completion of this thesis.

Lastly, I am incredibly thankful to my friends and colleagues, who have been by my side through both the good times and the challenges. Your presence and support have helped me grow, and for that, I am truly thankful.

This work was partially supported by the Portuguese Fundação para a Ciência e a Tecnologia (FCT) through project FirePuma (<https://doi.org/10.54499/PCIF/MPG/0156/2019>), through LARSyS FCT funding (DOI: 10.54499/LA/P/0083/2020, 10.54499/UIDP/50009/2020, and 10.54499/UIDB/50009/2020) and through COPELABS, University Lusófona project 10.54499/UIDB/ 04111/2020.



# Abstract

This thesis addresses the challenges of State Estimation (SE) for linear systems under diverse noise conditions, using both single and multi-sensor architectures. Traditional SE methods often assume Gaussian noise, which can limit accuracy when this assumption does not hold. This work explores novel filtering methods, leveraging Characteristic Functions (CFs) to handle non-Gaussian noise effectively. We also investigate the Wasserstein Barycenter (WB) method. The WB-based approach addresses challenges in integrating sensor data with diverse and inconsistent noise patterns, providing a more reliable solution for SE.

We propose and analyze two hybrid approaches for SE, combining the strengths of CF filtering and particle-based methods to address limitations of traditional Gaussian assumptions. The first uses a Dirac-based approximation to discretize the CF, while the second employs Gaussian mixtures to approximate the CF. Both approaches are evaluated under various noise scenarios, including Gaussian, Gaussian mixture, and exponential noise, for single-sensor configurations. The proposed methods are benchmarked against classical techniques such as the Kalman Filter (KF) and Particle Filter (PF). Simulation results demonstrate that the hybrid CF-based filters offer robustness and accuracy in non-Gaussian environments.

The second part of this thesis explores using WB for sensor fusion in multi-sensor systems. The WB approach improves SE accuracy and reliability by integrating data from sensors with diverse noise characteristics. This method provides a robust way to combine measurements while minimizing the impact of inconsistent noise. The efficacy of the WB-based approach is demonstrated through simulations,

highlighting its advantages in scenarios with heterogeneous sensor noise.

## **Keywords**

State Estimation; Non-Gaussian Noise; Bayes Filter; Wasserstein Barycenter; Multi-Agent Systems; Kalman Filter; Characteristic Function Filter.

# Resumo

A presente dissertação aborda os desafios da estimação de estado para sistemas lineares sob diversas condições de ruído, utilizando arquiteturas de um único sensor e multi-sensor. Os métodos tradicionais de estimação de estado frequentemente assumem ruído Gaussiano, o que pode limitar a precisão quando essa suposição falha. Este trabalho explora novos métodos de filtragem, usando funções características para lidar de forma eficaz com ruídos não-Gaussianos. Também foi investigado um método baseado no baricentro de Wasserstein. Esta abordagem pretende resolver desafios na integração de dados de sensores com padrões de ruído diversos e inconsistentes, proporcionando uma solução mais confiável para estimação de estado.

Propomos e analisamos duas abordagens híbridas para estimação de estado, combinando as vantagens da filtragem por função característica e métodos de partículas para abordar as limitações das suposições Gaussianas tradicionais. A primeira utiliza uma aproximação baseada em Diracs para discretizar a função característica, enquanto a segunda emprega misturas de Gaussianas para aproximar a função característica. Ambas as abordagens são avaliadas em vários cenários de ruído, incluindo ruído Gaussiano, mistura Gaussiana e ruído exponencial, em configurações de sensor único. Os métodos propostos são comparados com técnicas clássicas, como o filtro de Kalman e o filtro de partículas. Os resultados das simulações demonstram que os filtros híbridos baseados em funções características oferecem robustez e precisão em ambientes com ruído não-Gaussiano.

A segunda parte desta tese explora o uso do baricentro de Wasserstein para fusão de sensores em sistemas multi-sensores. A abordagem baseada no baricentro de Wasserstein melhora a precisão e confiabilidade da estimação de estado ao integrar dados de sensores com características de ruído diversas, minimizando as inconsistências. As simulações validam a eficácia desta abordagem, desta-

cando suas vantagens em cenários com ruído heterogêneo.

## **Palavras Chave**

Estimação de Estado; Ruído Não Gaussiano; Filtro de Bayes; Baricentro de Wasserstein; Sistemas Multi-Agente; Filtro de Kalman; Filtro de Função Característica.

# Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Introduction</b>  | <b>1</b> |
| 1.1      | Motivation . . . . .                                       | 2        |
| 1.2      | Objectives . . . . .                                       | 3        |
| 1.3      | Thesis Outline . . . . .                                   | 3        |
| <b>2</b> | <b>Single Sensor State Estimation</b>                      | <b>5</b> |
| 2.1      | Related Work . . . . .                                     | 6        |
| 2.1.1    | Bayes Filter . . . . .                                     | 6        |
| 2.1.2    | Single Sensor Gaussian Assumption . . . . .                | 7        |
| 2.1.2.A  | Kalman Filter . . . . .                                    | 7        |
| 2.1.3    | Single Sensor Non-Gaussian Assumption . . . . .            | 7        |
| 2.1.3.A  | Particle Filter . . . . .                                  | 8        |
| 2.1.3.B  | Gaussian Mixture Filter . . . . .                          | 8        |
| 2.1.3.C  | Characteristic Function Filter . . . . .                   | 9        |
| 2.2      | Single Sensor Problem . . . . .                            | 10       |
| 2.3      | Background Technical Details . . . . .                     | 11       |
| 2.3.1    | Bayes Filter . . . . .                                     | 11       |
| 2.3.2    | Single Sensor Gaussian Assumption . . . . .                | 12       |
| 2.3.2.A  | Kalman Filter . . . . .                                    | 12       |
| 2.3.3    | Single Sensor Non-Gaussian Assumption . . . . .            | 13       |
| 2.3.3.A  | Particle Filter . . . . .                                  | 13       |
| 2.3.3.B  | Gaussian Mixture Filter . . . . .                          | 14       |
| 2.3.3.C  | Characteristic Function Filter . . . . .                   | 17       |
| 2.4      | Proposed Approach . . . . .                                | 18       |
| 2.4.1    | Hybrid Filter . . . . .                                    | 18       |
| 2.4.2    | Hybrid Filter via Dirac Approximation . . . . .            | 19       |
| 2.4.2.A  | Examples . . . . .   | 22       |
| 2.4.3    | Hybrid Filter via Gaussian Mixture Approximation . . . . . | 25       |

|          |  |           |
|----------|--|-----------|
| 2.5      | Experimental Results and Discussion . . . . .                            | 27        |
| 2.5.1    | Simulation . . . . .   | 27        |
| 2.5.2    | Results . . . . .  | 28        |
| 2.5.2.A  | Hybrid Filter via Dirac Approximation . . . . .                          | 28        |
| 2.5.2.B  | Hybrid Filter via Gaussian Mixture Approximation . . . . .               | 36        |
| 2.5.2.C  | Approaches Comparison . . . . .  | 41        |
| <b>3</b> | <b>Multi-Sensor State Estimation</b>                                     | <b>42</b> |
| 3.1      | Related Work . . . . .   | 43        |
| 3.1.1    | Multi-Sensor Gaussian Assumption . . . . .                               | 43        |
| 3.1.1.A  | Distributed Sensor Fusion Based on Average Consensus . . . . .           | 43        |
| 3.1.1.B  | Distributed Kalman Filtering . . . . .                                   | 44        |
| 3.1.2    | Multi-Sensor Non-Gaussian Assumption . . . . .                           | 45        |
| 3.1.2.A  | Distributed State Estimation Based on Kullback–Leibler Average . . . . . | 45        |
| 3.2      | Multi-Sensor Problem . . . . .   | 46        |
| 3.3      | Background Technical Details . . . . .                                   | 47        |
| 3.3.1    | Multi-Sensor Gaussian Assumption . . . . .                               | 47        |
| 3.3.1.A  | Distributed Sensor Fusion Based on Average Consensus . . . . .           | 47        |
| 3.3.1.B  | Distributed Kalman Filtering . . . . .                                   | 48        |
| 3.3.2    | Multi-Sensor Non-Gaussian Assumption . . . . .                           | 49        |
| 3.3.2.A  | Distributed State Estimation Based on Kullback–Leibler Average . . . . . | 49        |
| 3.3.3    | Wasserstein Barycenter . . . . .   | 50        |
| 3.4      | Proposed Approach . . . . .  | 51        |
| 3.4.1    | Wasserstein Barycenter for Sensor Fusion . . . . .                       | 51        |
| 3.5      | Experimental Results and Discussion . . . . .                            | 53        |
| 3.5.1    | Simulation . . . . .   | 54        |
| 3.5.2    | Results . . . . .  | 56        |
| 3.5.2.A  | Gaussian Noises Simulation . . . . .                                     | 56        |
| 3.5.2.B  | Different Noises Simulation . . . . .                                    | 58        |
| <b>4</b> | <b>Conclusion</b>  | <b>60</b> |
| 4.1      | Summary . . . . .  | 61        |
| 4.2      | Future Work . . . . .  | 61        |
|          | <b>Bibliography</b>  | <b>63</b> |

# List of Figures

|      |  |    |
|------|--|----|
| 2.1  | Gaussian <i>PDF</i> , approximation on the left and real <i>PDF</i> on the right. . . . .  | 22 |
| 2.2  | Real and approximated <i>CF</i> 's of gaussian random variable. . . . .  | 23 |
| 2.3  | Gaussian Mixture <i>PDF</i> , approximation on the left and real <i>PDF</i> on the right. . . . .  | 23 |
| 2.4  | Real and approximated <i>CF</i> 's of gaussian mixture random variable. . . . .  | 24 |
| 2.5  | Exponential <i>PDF</i> , approximation on the left and real <i>PDF</i> on the right. . . . .   | 24 |
| 2.6  | Real and approximated <i>CF</i> 's of exponential random variable. . . . .   | 25 |
| 2.7  | Cumulative Root Mean Square Error (RMSE) of the position of Gaussian walk with Gaussian noise: a) Characteristic Function (CF) Filter with 256 particles; b) Kalman Filter; c) Particle Filter with 256 particles; d) Average of the cumulative RMSE over the 50 runs. . . . . | 29 |
| 2.8  | Covariance Matrix trace of the position estimate uncertainty in the Gaussian walk with Gaussian noise: a) CF Filter with 256 particles; b) Kalman Filter; c) Particle Filter with 256 particles. . . . .   | 30 |
| 2.9  | Cumulative RMSE of the position of Gaussian walk with Gaussian noise: a) CF Filter with 1296 particles; b) Particle Filter with 1296 particles; c) Average of the cumulative RMSE over the 50 runs. . . . .  | 31 |
| 2.10 | Covariance Matrix trace of the position estimate uncertainty in the Gaussian walk with Gaussian noise: a) CF Filter with 1296 particles; b) Particle Filter with 1296 particles. . . . .   | 32 |
| 2.11 | Cumulative RMSE of the position of Gaussian Mixture walk with Exponential noise: a) CF Filter with 1296 particles; b) Kalman Filter; c) Particle Filter with 1296 particles; d) Average of the cumulative RMSE over the 50 runs. . . . .                                       | 33 |
| 2.12 | Cumulative RMSE of the position of Gaussian Mixture w/ 2 peaks walk with Gaussian Mixture noise: a) CF Filter with 3 Gaussians; b) Kalman Filter; c) Particle Filter with 256 particles; d) Average of the cumulative RMSE over the 50 runs. . . . .                           | 37 |
| 2.13 | Cumulative RMSE of the position of Gaussian Mixture w/ 2 peaks walk with Gaussian Mixture noise: a) CF Filter with 4 Gaussians; b) Average of the cumulative RMSE over the 50 runs. . . . .  | 38 |

|      |   |    |
|------|---|----|
| 2.14 | Covariance Matrix trace of the position estimate uncertainty in the Gaussian Mixture w/ 2 peaks walk with Gaussian Mixture noise: a) CF Filter with 3 Gaussians; b) CF Filter with 4 Gaussians. . . . . | 38 |
| 3.1  | Wasserstein Barycenter along increasing iterations. . . . .   | 53 |
| 3.2  | Sensor network example. . . . .   | 55 |
| 3.3  | Information fusion diagram. . . . .   | 55 |
| 3.4  | Gaussian noises simulation: a) $\ell^2$ -norm of the position error; b) Covariance matrix trace of the position uncertainty; c) RMSE over multiple simulations. . . . .                                 | 57 |
| 3.5  | Different noises simulation: a) $\ell^2$ -norm of the position error; b) Covariance matrix trace of the position uncertainty; c) RMSE over multiple simulations. . . . .                                | 59 |

# List of Tables

|     |  |    |
|-----|--|----|
| 2.1 | RMSE of all filters along different simulation scenarios, Dirac approach. . . . .          | 35 |
| 2.2 | RMSE of all filters along different simulation scenarios, Gaussian Mixture approach. . . . | 40 |
| 3.1 | RMSE of Gaussian noise simulation. . . . .   | 57 |
| 3.2 | RMSE of different noises simulation. . . . .   | 59 |

# Acronyms

|               |                                      |
|---------------|--------------------------------------|
| <b>CF</b>     | Characteristic Function              |
| <b>DKF</b>    | distributed Kalman filtering         |
| <b>EKF</b>    | Extended Kalman Filter               |
| <b>GMF</b>    | Gaussian Mixture Filter              |
| <b>GMM</b>    | Gaussian Mixture Model               |
| <b>LKF</b>    | local Kalman filtering               |
| <b>KF</b>     | Kalman Filter                        |
| <b>KL</b>     | Kullback–Leibler                     |
| <b>MA</b>     | Multi-Agent                          |
| <b>MA-GA</b>  | Multi-Agent Gaussian Assumption      |
| <b>MA-NGA</b> | Multi-Agent Non-Gaussian Assumption  |
| <b>PF</b>     | Particle Filter                      |
| <b>PDF</b>    | Probability Density Function         |
| <b>RMSE</b>   | Root Mean Square Error               |
| <b>SA</b>     | Single Agent                         |
| <b>SA-GA</b>  | Single Agent Gaussian Assumption     |
| <b>SA-NGA</b> | Single Agent Non-Gaussian Assumption |
| <b>SE</b>     | State Estimation                     |
| <b>UKF</b>    | Unscented Kalman Filter              |
| <b>WB</b>     | Wasserstein Barycenter               |

# 1

## Introduction

### Contents

---

|                              |   |
|------------------------------|---|
| 1.1 Motivation . . . . .     | 2 |
| 1.2 Objectives . . . . .     | 3 |
| 1.3 Thesis Outline . . . . . | 3 |

---

## 1.1 Motivation

State Estimation (SE) is a fundamental concept with broad applications across various engineering disciplines. In control systems, it is pivotal for feedback mechanisms that ensure system stability and optimal performance. In robotics, SE is integral for critical tasks such as localization, mapping, and navigation. Similarly, in signal processing, SE plays an essential role in noise filtering and the reconstruction of signals from partial or corrupted data. While the concept is well-established, the increasing complexity of modern systems, alongside advancements in big data and the Internet of Things, has elevated the importance of SE. The growing proliferation of sensors has further amplified the demand for robust and precise SE algorithms. For these algorithms to be applicable in real-time systems, they must also demonstrate computational efficiency.

Traditional SE methods often rest on assumptions that may not hold in practical scenarios. For instance, many established techniques assume Gaussian noise for mathematical tractability, even though this assumption may not accurately reflect the actual noise characteristics. The Kalman Filter (KF) [1] and its variants have long been the bedrock for SE in linear systems, offering optimal solutions under the Gaussian noise assumption. However, these methods can experience significant degradation in estimation accuracy under non-Gaussian noise conditions. The Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) [2, 3] were introduced to extend the applicability to nonlinear systems, but they still depend on Gaussian noise assumptions, which might limit their efficacy in non-Gaussian environments.

Particle Filters (PFs) [4], which explicitly model non-Gaussian and nonlinear systems through sampling, have emerged as powerful alternatives. Nevertheless, PFs encounter significant issues, such as the curse of dimensionality and substantial computational costs, limiting their use in real-time applications with constrained resources.

Furthermore, most traditional methods are primarily designed for single-sensor systems and are often extended to centralized architectures where multiple sensors are integrated into a unified data source for SE. However, centralized architectures are prone to single points of failure and may be impractical in settings where sensors are geographically distributed. Distributed and decentralized sensor networks provide a more resilient and scalable solution, but they introduce additional challenges in data fusion and consensus algorithms.

The primary objective of this thesis is to explore the complexities and challenges of SE in linear systems. This research problem is multifaceted and can be categorized based on two critical dimensions: the type of noise affecting the system and the architecture used for SE. Specifically, the architecture can be classified as either a Single Agent (SA) system, which typically involves a single sensor, or a Multi-Agent (MA) system, which incorporates a network of sensors. The noise can be Gaussian or non-Gaussian, resulting in four categories: Single Agent Gaussian Assumption (SA-GA), Single Agent Non-Gaussian Assumption (SA-NGA), Multi-Agent Gaussian Assumption (MA-GA), and

Multi-Agent Non-Gaussian Assumption (MA-NGA).

In scenarios involving a single sensor with Gaussian noise (SA-GA), techniques such as the Kalman Filter [1] have historically been the preferred approach. However, these techniques may not be robust against deviations from Gaussian noise, leading to performance degradation when the noise characteristics significantly diverge from Gaussian. When dealing with non-Gaussian noise in single-sensor systems (SA-NGA), alternative approaches such as PFs [4], Characteristic Function (CF) Filters [5–7], among others [8], are employed. While these methods can handle non-Gaussian noise, they tend to be computationally demanding and may not be feasible for real-time applications.

Moving from single-sensor to multi-sensor systems adds new layers of complexity. In multi-sensor systems with Gaussian noise (MA-GA), the problem extends beyond SE to include the challenge of fusing data from multiple sensors. Centralized methods can be applied in these scenarios, but they may lack efficiency and robustness, particularly in distributed or decentralized architectures ([9–13]). Finally, in multi-sensor systems with non-Gaussian noise (MA-NGA), the SE problem becomes substantially more complex. The challenge of fusing data from multiple sensors is further worsened by the presence of non-Gaussian noise characteristics [14–16].

## 1.2 Objectives

The primary objectives of this thesis are to address key challenges in SE by:

- **Characteristic Functions in Filters:** Investigate the potential of CFs to enhance the performance and robustness of filtering algorithms, particularly in scenarios where traditional assumptions, such as Gaussian noise, do not hold.
- **Wasserstein Barycenter for Sensor Fusion:** Explore the use of the Wasserstein Barycenter (WB) as an innovative approach for sensor fusion, with the aim of improving the accuracy and reliability of SE in multi-sensor systems, especially when sensors exhibit diverse noise characteristics.

## 1.3 Thesis Outline

The remainder of this thesis is structured as follows:

- **Chapter 2:** This chapter addresses the challenge of SE in linear systems using a single sensor, with a focus on both Gaussian and non-Gaussian noise assumptions. The chapter explores foundational filtering techniques such as the KF, PF, and CF Filter. Special emphasis is placed on the theoretical derivations of these methods, covering the prediction and update steps essential

for Bayesian filtering frameworks. Lastly, a novel Hybrid Filter is derived and simulation results are presented to evaluate its performance under various noise distributions, including Gaussian and non-Gaussian scenarios;

- **Chapter 3:** This chapter focuses on the challenges and methodologies associated with state estimation using a network of multiple sensors. It begins by examining multi-sensor SE under the Gaussian noise assumption, introducing distributed sensor fusion techniques based on average consensus and distributed Kalman filtering (DKF). Furthermore, it explores state estimation under non-Gaussian noise assumptions, with methods such as Distributed State Estimation based on the Kullback–Leibler (KL) average. Finally, the chapter delves into advanced sensor fusion techniques using the WB framework, emphasizing its applicability in centralized sensor networks;
- **Chapter 4:** The final chapter summarizes the key contributions of the thesis and proposes directions for future research.

# 2

## Single Sensor State Estimation

### Contents

---

|     |   |    |
|-----|---|----|
| 2.1 | Related Work . . . . .                        | 6  |
| 2.2 | Single Sensor Problem . . . . .               | 10 |
| 2.3 | Background Technical Details . . . . .        | 11 |
| 2.4 | Proposed Approach . . . . .                   | 18 |
| 2.5 | Experimental Results and Discussion . . . . . | 27 |

---

Chapter 2 provides an in-depth exploration of the challenges and methodologies associated with SE in linear systems utilizing a single sensor. The primary objective is to estimate the system state based on noisy observations, considering both Gaussian and non-Gaussian noise environments. Various filtering techniques, such as the KF, PF, and the CF Filter, are thoroughly examined. The efficacy of these approaches is demonstrated through rigorous theoretical formulations complemented by simulation results.

## 2.1 Related Work

This section aims to provide a comprehensive overview of the existing literature and methodologies in the field of single sensor SE. It also investigates the practical applications of CFs in filtering.

### 2.1.1 Bayes Filter

To address the problem of SE within linear systems, it is essential to first discuss the fundamental framework that enables optimal filtering: the Bayes Filter. This filter forms the foundational basis for both SA and MA state estimation under Gaussian and non-Gaussian noise assumptions.

Bayesian filtering revolves around the probabilistic interpretation of states based on the available data. Given a system model and a sequence of observations, the objective is to estimate the sequence of states that is most likely to have produced the observed data. The Bayes Filter's formulation is structured around two fundamental steps: prediction and update.

- **Prediction Step:** In this step, the current belief regarding the system's state is extended into the future based on the system model. It involves forecasting the state at the next time step, taking into account the known dynamics of the system as well as prior knowledge of the state.
- **Update Step:** This step incorporates new observational data into the predicted state, refining the estimate based on actual measurements. The state estimate is adjusted by comparing the newly observed data to the predicted values, thereby improving the accuracy of the state estimate.

The Bayes Filter is inherently recursive, continually refining predictions and updates as new information becomes available. It is also crucial to emphasize the Markov assumption, which serves as a foundational premise for the filter's operation. According to this assumption, the state at any given time  $t$  depends solely on the state at the previous time  $t - 1$ , and is conditionally independent of all preceding states.

## 2.1.2 Single Sensor Gaussian Assumption

In this subsection, we examine the problem of Single Sensor SE under the Gaussian Assumption (SA-GA), presenting one of the major contributions to this field.

### 2.1.2.A Kalman Filter

The KF [1] represents a seminal contribution to the field. Introduced by R.E. Kalman in 1960, the KF serves as the cornerstone for SE in linear systems affected by Gaussian noise. The algorithm provides an optimal solution under the assumptions of system linearity and Gaussian noise, balancing computational efficiency with estimation accuracy.

Like the Bayes Filter, the KF employs a two-step process: prediction and update. However, the KF leverages the linearity and Gaussian noise assumptions to simplify these steps into matrix operations, thereby significantly enhancing computational efficiency.

- **Prediction Step:** This step propagates the current state estimate forward in time using the system's model. This propagation accounts for any known control input and incorporates uncertainty in the process via the process noise covariance matrix. The outcome is an *a priori* estimate of the state at the next time step.
- **Update Step:** In this step, the *a priori* estimate is refined by incorporating the new measurement. This involves adjusting the predicted state based on the residual, which quantifies the difference between the predicted and observed values. The residual is then used to update the state estimate, with the Kalman Gain serving as a key component that determines the relative weight assigned to the new measurement versus the prediction.

The matrix operations used in the KF not only reduce computational complexity but also provide an optimal solution under the Gaussian noise assumption. However, it is important to note that while the KF delivers optimal solutions under conditions of linearity and Gaussian noise, it becomes vulnerable when these assumptions are violated. Extensions such as the EKF and UKF [2,3] have been developed to address nonlinearities while maintaining the Gaussian noise assumption.

## 2.1.3 Single Sensor Non-Gaussian Assumption

In the domain of SE for linear systems, the SA-NGA presents a unique set of challenges that are not adequately addressed by conventional Gaussian-based methods. Consequently, this subsection aims to explore advanced methodologies that effectively tackle these challenges.

### 2.1.3.A Particle Filter

The PF [4] offers a powerful approach for state estimation in systems characterized by nonlinear dynamics and non-Gaussian noise distributions. It approximates the posterior Probability Density Function (PDF) through a collection of weighted samples, commonly referred to as particles.

The filter operates through a sequence of well-defined steps, each contributing to the overall state estimation process:

- **Initialization:** The initial step involves generating a set of particles (samples) that represent possible states of the system. Each particle is considered a hypothesis of the true state and is initialized based on the prior distribution, which reflects the initial knowledge or belief about the system's state before any measurements are obtained.
- **Prediction:** Each particle is propagated forward in time according to the system model. This prediction step incorporates the system's process model, including any control inputs and the inherent uncertainties, leading to a diverse set of predicted states.
- **Weight Update and Normalization:** In this step, each particle is assigned a weight that quantifies how well it aligns with the newly acquired measurement data. The weight represents the likelihood of the observed data given the state hypothesis represented by the particle. These weights are then normalized to ensure that their sum equals one.
- **Resampling:** Resampling is a critical step to mitigate particle degeneracy, a situation where many particles acquire negligible weights over time. During resampling, particles with higher weights are more likely to be selected to form the next generation, while particles with lower weights are discarded.

Although PFs are highly versatile and capable of managing complex nonlinear and non-Gaussian systems, they come with substantial computational costs. Thus, there is an inherent trade-off between estimation accuracy and computational efficiency.

### 2.1.3.B Gaussian Mixture Filter

The Gaussian Mixture Filter (GMF) [8] extends Bayesian estimation principles to state-space systems operating under non-Gaussian noise conditions by utilizing Gaussian Mixture Models (GMMs).

The GMF operates through the familiar two-step prediction and update process. During the prediction step, the GMF estimates the state at the subsequent time step by applying system dynamics, while the update step refines this prediction based on the latest sensor measurements.

A fundamental aspect of the GMF is the Gaussian mixture reduction technique [17], which employs KL divergence to efficiently merge components of the Gaussian mixtures. This method ensures the

integrity of the statistical model while addressing computational challenges arising from the exponential growth of mixture components during the estimation process.

The prediction and update steps in GMF are conceptually analogous to those of the KF, but here they are applied to each component of the GMM. Following each prediction or update, a GMM reduction algorithm is employed to maintain computational tractability. This algorithm deterministically combines pairs of mixture components to form a reduced mixture, continuing iteratively until a predefined KL divergence threshold is surpassed, which signifies an acceptable level of information loss. This ensures that the essential features of the PDF are preserved while reducing the computational complexity.

### **2.1.3.C Characteristic Function Filter**

In the context of SE for linear systems, CFs emerge as a compelling tool due to their ability to simplify complex operations, such as convolution, by transforming them into straightforward multiplications in the frequency domain.

CFs are essentially the Fourier transform of PDFs, providing a representation in the frequency domain that encapsulates all the integral properties of the distribution. Their utility in filtering is particularly advantageous under non-Gaussian noise conditions, as CFs provide an alternative and more tractable representation of PDFs, enabling efficient execution of critical filtering operations like convolution.

The CF Filter employs the same two-step recursive process as the Bayesian framework, but with a key distinction: the use of CFs instead of PDFs. Foundational work on this method is documented in [5, 6], with a comprehensive formulation provided in [5]. In the prediction step within this transformed domain, the process is reduced to a multiplication of the CF representing the current belief (prior) and the CF characterizing the transition model. The transition model's CF arises solely from the disturbances acting on the system.

Conversely, the update step takes the form of a convolution, suggesting that the CF Filter is conceptually similar to the Bayes Filter, with the convolution operation occurring in the update step rather than in the prediction step as in the Bayes Filter.

The application presented in [6] addresses scenarios where only state propagation via the transition model is necessary, i.e., only the prediction step is performed. This approach is beneficial in cases where there are no observational data available, as the prediction step is merely a multiplication operation in the frequency domain.

To manage the convolution during the update step, the framework introduced in [7] is employed. Following the prediction step, the inverse fast Fourier transform is used to revert back to the time domain, facilitating the update step. Subsequently, the fast Fourier transform is reapplied to return to the frequency domain for the subsequent prediction. This iterative process allows the CF Filter to perform recursive state estimation with enhanced computational efficiency.

## 2.2 Single Sensor Problem

The purpose of this section is to present a rigorous mathematical formulation of the state estimation problem addressed in this work from the perspective of a single sensor.

The task of SE in a single-sensor context, here termed the single sensor problem, involves estimating the state of a dynamic system based on observations collected by a solitary sensor. Specifically, the objective is to accurately infer the system state, denoted as  $x_t$ , at each time step given a sequence of observations  $z_1, z_2, \dots, z_t$  obtained from the sensor. The fundamental aim is to construct an estimator that can effectively handle noisy, and potentially incomplete, observations to yield the most accurate estimate,  $\hat{x}_t$ , of the true system state.

This problem is formulated using the following state-space model, which comprises two fundamental equations: the state equation and the observation equation.

- **State Equation:** Equation 2.1 describes the temporal evolution of the state. Here,  $x_t \in \mathbb{R}^n$  represents the state vector at time  $t$ , encapsulating the true underlying state of the system. The matrix  $A \in \mathbb{R}^{n \times n}$  is the state transition matrix that governs the evolution of the state between consecutive time steps. The matrix  $B \in \mathbb{R}^{n \times d}$  is the control input matrix, which maps the control input  $u_t$  to the state space. The control input vector  $u_t \in \mathbb{R}^d$  represents any external influences at time  $t$ . Lastly,  $d_t \sim \mathcal{D}(\alpha)$  denotes the process noise vector, where  $\mathcal{D}(\alpha)$  represents a general distribution  $\mathcal{D}$  parameterized by  $\alpha$ .

$$x_t = Ax_{t-1} + Bu_t + d_t \quad (2.1)$$

- **Observation Equation:** Equation 2.2 relates the state of the system to the measurements collected by the sensor. In this equation,  $z_t \in \mathbb{R}^p$  represents the observation or measurement vector at time  $t$ , encapsulating the information obtained by the sensor. The matrix  $H \in \mathbb{R}^{p \times n}$  is the observation matrix, which maps the state space to the observation space. The measurement noise vector  $v_t \sim \mathcal{D}(\theta)$ , analogous to the process noise, is drawn from a distribution  $\mathcal{D}$  parameterized by  $\theta$ .

$$z_t = Hx_t + v_t \quad (2.2)$$

The problem setup operates under several key assumptions. First, linearity is assumed in both the state and observation equations, ensuring that each equation is linear with respect to its respective variables. Second, the system is assumed to be time-invariant, meaning that the matrices  $A$ ,  $B$ , and  $H$  are constant over time. Additionally, these matrices are assumed to be known *a priori*, and the pair  $(A, H)$  is considered to be observable. Lastly, the noise vectors  $d_t$  and  $v_t$  are presumed to be mutually independent and drawn from known distributions.

## 2.3 Background Technical Details

Section 2.3 delves into the technical specifics and mathematical formulations underpinning the methodologies discussed in Section 2.1.

### 2.3.1 Bayes Filter

As previously introduced, the Bayesian Filter is formulated around two fundamental steps: prediction and update.

- **Prediction Step:** This step propagates the prior belief about the system's state forward in time, integrating the system's dynamics and the associated uncertainties. Mathematically, it is represented as:

$$p(x_t|z_{1:t-1}) = \int p(x_t|x_{t-1}) \cdot p(x_{t-1}|z_{1:t-1}) dx_{t-1} \quad (2.3)$$

Here,  $x_t$  denotes the state at time  $t$ ,  $z_{1:t-1}$  represents the sequence of observations up to time  $t-1$ ,  $p(x_t|x_{t-1})$  is the transition model that describes the system's dynamics, and  $p(x_{t-1}|z_{1:t-1})$  is the prior state estimate at time  $t-1$ .

- **Update Step:** This step refines the prediction by incorporating the latest observation at time  $t$ . It adjusts the state probability based on the likelihood of the new observation given the predicted state:

$$p(x_t|z_{1:t}) = \frac{p(z_t|x_t) \cdot p(x_t|z_{1:t-1})}{p(z_t|z_{1:t-1})} \quad (2.4)$$

In this expression,  $z_t$  is the new observation at time  $t$ ,  $p(z_t|x_t)$  represents the likelihood of the observation given the current state, and  $p(z_t|z_{1:t-1})$  serves as the normalization factor, ensuring that the posterior distribution remains a valid probability distribution.

The Markov assumption, previously discussed as a key premise, can be expressed mathematically as:

$$p(x_t|x_{t-1}, x_{t-2}, \dots, x_0) = p(x_t|x_{t-1}) \quad (2.5)$$

where  $x_t$  denotes the state at time  $t$ .

## 2.3.2 Single Sensor Gaussian Assumption

### 2.3.2.A Kalman Filter

The KF operates on the principle of recursive Bayesian estimation, employing a two-step process: prediction and update. The mathematical formulation of these steps is as follows:

- **Prediction Step:** Let  $\hat{x}_{t|t-1} \in \mathbb{R}^n$  denote the *a priori* state estimate at time  $t$ , based on all available information up to time  $t - 1$ . The matrix  $A \in \mathbb{R}^{n \times n}$  is the state transition matrix, and  $\hat{x}_{t-1|t-1} \in \mathbb{R}^n$  is the *a posteriori* state estimate at time  $t - 1$ . The matrix  $B \in \mathbb{R}^{n \times m}$  is the control input matrix, and  $u_t \in \mathbb{R}^d$  represents the control input at time  $t$ . Furthermore,  $P_{t|t-1} \in \mathbb{R}^{n \times n}$  is the *a priori* error covariance matrix, and  $Q \in \mathbb{R}^{n \times n}$  is the process noise covariance matrix. The prediction equations are given by:

$$\hat{x}_{t|t-1} = A\hat{x}_{t-1|t-1} + Bu_t \quad (2.6)$$

$$P_{t|t-1} = AP_{t-1|t-1}A^\top + Q. \quad (2.7)$$

These equations propagate the current state estimate and its associated uncertainty forward in time, incorporating system dynamics and the effects of control inputs.

- **Update Step:** Define the following variables:  $\hat{x}_{t|t} \in \mathbb{R}^n$  as the *a posteriori* state estimate at time  $t$ ,  $K_t \in \mathbb{R}^{n \times p}$  as the Kalman gain,  $z_t \in \mathbb{R}^p$  as the measurement at time  $t$ , and  $H \in \mathbb{R}^{p \times n}$  as the observation model matrix. The measurement noise covariance is denoted by  $R \in \mathbb{R}^{p \times p}$ , and  $P_{t|t} \in \mathbb{R}^{n \times n}$  is the *a posteriori* error covariance matrix. The update equations are:

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(z_t - H\hat{x}_{t|t-1}) \quad (2.8)$$

$$K_t = P_{t|t-1}H^\top(HP_{t|t-1}H^\top + R)^{-1} \quad (2.9)$$

$$P_{t|t} = (I - K_tH)P_{t|t-1} \quad (2.10)$$

The Kalman gain  $K_t$  determines the weight given to the new measurement in updating the state estimate. The equations adjust the predicted state estimate and error covariance to refine the estimate based on the latest available measurement.

In these equations,  $n$  represents the state dimension,  $m$  denotes the dimensionality of the control input,  $p$  is the dimensionality of the measurement, and  $I$  is the identity matrix of appropriate dimensions.

## 2.3.3 Single Sensor Non-Gaussian Assumption

### 2.3.3.A Particle Filter

The PF provides an effective solution for state estimation in non-Gaussian and nonlinear contexts by employing a sequence of steps, each contributing uniquely to the estimation process. Let  $x_t^{(i)} \in \mathbb{R}^n$  represent the state of particle  $i$  at time  $t$ ,  $u_t \in \mathbb{R}^d$  the control input at time  $t$ ,  $z_t \in \mathbb{R}^p$  the measurement at time  $t$ , and  $w_t^{(i)} \in \mathbb{R}$  the weights reflecting the importance of each particle in representing the posterior distribution.

- **Initialization:** Initialize  $N$  particles  $\{x_0^{(i)}\}$  from the prior distribution  $p(x_0)$ . Each particle  $x_0^{(i)}$  is assigned an initial weight  $w_0^{(i)}$ , typically set to  $\frac{1}{N}$ :

$$x_0^{(i)} \sim p(x_0), \quad i = 1, \dots, N \quad (2.11)$$

- **Prediction:** Predict the next state for each particle using the process model:

$$x_t^{(i)} \sim p(x_t | x_{t-1}^{(i)}, u_t) \quad (2.12)$$

This step advances each particle according to the system's dynamics, incorporating control inputs and process noise.

- **Weight Update and Normalization:** Upon receiving a new measurement  $z_t$ , the weight  $w_t^{(i)}$  of each particle is updated based on the likelihood  $p(z_t | x_t^{(i)})$ . Weights are subsequently normalized such that their sum equals one:

$$w_{t,\text{unnorm}}^{(i)} = w_{t-1}^{(i)} p(z_t | x_t^{(i)}) \quad (2.13)$$

$$w_t^{(i)} = \frac{w_{t,\text{unnorm}}^{(i)}}{\sum_{j=1}^N w_{t,\text{unnorm}}^{(j)}} \quad (2.14)$$

This step evaluates how well each particle explains the observed data and adjusts the weights accordingly.

- **Resampling:** Resample  $N$  particles  $\{x_t^{(i)'}\}$  based on the normalized weights  $\{w_t^{(i)}\}$ :

$$x_t^{(i)'} \sim \sum_{j=1}^N w_t^{(j)} \delta(x_t^{(i)'} - x_t^{(j)}) \quad (2.15)$$

where  $\delta(\cdot)$  is the Dirac delta function.

Resampling addresses particle degeneracy by favoring particles with higher weights, thereby ensuring an adequate representation of the posterior distribution.

### 2.3.3.B Gaussian Mixture Filter

The GMF leverages GMMs to estimate the state of a system from noisy observations. This estimation process is executed through iterative prediction and update steps, applied to each component of the GMM.

Considering the Bayesian filtering framework and state-space systems represented by a GMM, the prior distribution can be described as follows:

$$p(x_0) = \sum_{i=1}^{N_p} \alpha_i \mathcal{N}(x_0; \mu_0^i, P_0^i), \quad \sum_{i=1}^{N_p} \alpha_i = 1. \quad (2.16)$$

The process model is expressed in the form:

$$p(x_t | x_{t-1}) = \sum_{j=1}^{N_x} \beta_t^j \mathcal{N}(x_t; Ax_{t-1} + u_t^j, Q_t^j), \quad \sum_{j=1}^{N_x} \beta_t^j = 1. \quad (2.17)$$

The measurement model is similarly represented as:

$$p(z_t | x_t) = \sum_{k=1}^{N_z} \gamma_t^k \mathcal{N}(z_t; Cx_t + v_t^k, R_t^k), \quad \sum_{k=1}^{N_z} \gamma_t^k = 1. \quad (2.18)$$

In the above equations,  $\mathcal{N}(x; \mu, P)$  denotes a standard multivariate Gaussian distribution. The mean offset terms  $u_t^j$  and  $v_t^k$  facilitate the inclusion of input signals, such as control inputs or noise.

Assuming we have a predicted mixture available, it can be expressed as:

$$p(x_t | z_{1:t-1}) = \sum_{\ell=1}^{N_{t|t-1}} w_{t|t-1}^\ell \mathcal{N}(x_t; \hat{x}_{t|t-1}^\ell, P_{t|t-1}^\ell), \quad (2.19)$$

$$\sum_{\ell=1}^{N_{t|t-1}} w_{t|t-1}^\ell = 1.$$

Note that the prior distribution in Equation 2.16 is already in this form at  $t = 1$ . Now, considering the measurement update step given in Equation 2.4, this can be extended into the current GMM context as follows:

$$p(x_t | z_{1:t}) = \sum_{\ell=1}^{N_{t|t-1}} \sum_{k=1}^{N_z} w_{t|t-1}^\ell \gamma_t^k \frac{\mathcal{N}(z_t; Cx_t + v_t^k, R_t^k)}{p(z_t | z_{1:t-1})} \cdot \mathcal{N}(x_t; \hat{x}_{t|t-1}^\ell, P_{t|t-1}^\ell). \quad (2.20)$$

Due to the linear Gaussian structure, this expression can be reformulated as:

$$p(x_t | z_{1:t}) = \sum_{s=1}^{N_{t|t}} w_{t|t}^s \mathcal{N}(x_t; \hat{x}_{t|t}^s, P_{t|t}^s), \quad \sum_{s=1}^{N_{t|t}} w_{t|t}^s = 1, \quad (2.21)$$

where for each  $k = 1, \dots, N_z$  and  $\ell = 1, \dots, N_{t|t-1}$ , it holds that:

$$N_{t|t} = N_{t|t-1} N_z \quad (2.22)$$

$$s \triangleq N_z(\ell - 1) + k \quad (2.23)$$

$$\hat{x}_{t|t}^s = \hat{x}_{t|t-1}^\ell + K_t^s e_t^s \quad (2.24)$$

$$e_t^s = z_t - C \hat{x}_{t|t-1}^\ell - v_t^k \quad (2.25)$$

$$\Sigma_t^s = C P_{t|t-1}^\ell (C)^\top + R_t^k \quad (2.26)$$

$$K_t^s = P_{t|t-1}^\ell C (\Sigma_t^s)^{-1} \quad (2.27)$$

$$P_{t|t}^s = P_{t|t-1}^\ell - K_t^s \Sigma_t^s (K_t^s)^\top \quad (2.28)$$

$$w_{t|t}^s = \frac{\bar{w}_{t|t}^s}{\sum_{s=1}^{N_{t|t}} \bar{w}_{t|t}^s}, \quad (2.29)$$

$$\bar{w}_{t|t}^s = w_{t|t-1}^\ell \gamma_t^k \quad (2.30)$$

With this filtering distribution established, we can proceed to the prediction step as described in Equation 2.3, which is expressed in the GMM context as:

$$p(x_t | z_{1:t}) = \sum_{s=1}^{N_{t|t}} \sum_{j=1}^{N_x} w_{t|t}^s \beta_t^j \int \mathcal{N}(x_t; Ax_t + u_t^j, Q_t^j) \mathcal{N}(x_t; \hat{x}_{t|t}^s, P_{t|t}^s) dx_t \quad (2.31)$$

again, due to the linear Gaussian densities involved, we can express this predicted mixture via

$$p(x_t | z_1) = \sum_{\ell=1}^{N_{t+1|t}} w_{t+1|t}^\ell \mathcal{N}(x_t; \hat{x}_{t+1|t}^\ell, P_{t+1|t}^\ell), \quad \sum_{\ell=1}^{N_{t+1|t}} w_{t+1|t}^\ell = 1, \quad (2.32)$$

where for each  $s = 1, \dots, N_{t|t}$  and  $j = 1, \dots, N_x$ , we have:

$$N_{t+1|t} = N_{t|t}N_x, \quad (2.33)$$

$$\ell \triangleq N_x(s-1) + j, \quad (2.34)$$

$$\hat{x}_{t+1} | t^\ell = A\hat{x}_t | t^s + u_t^j, \quad (2.35)$$

$$P_{t+1|t}^\ell = AP_{t|t}^s A^\top + Q_t^j, \quad (2.36)$$

$$w_{t+1|t}^\ell = w_{t|t}^s \beta_t^j. \quad (2.37)$$

Thus, we return to the initial assumption given in Equation 2.19, now at one time step ahead, enabling the recursive procedure to continue.

Upon completing the prediction and update steps, the GMM reduction algorithm is employed to merge components and maintain manageable model complexity. The main goal of this reduction process is to reduce a GMM of the form:

$$\pi(x) = \sum_{i=1}^N w_i \pi_i(x), \quad \pi_i = \mathcal{N}(x; \mu_i, P_i), \quad \sum_{i=1}^N w_i = 1, \quad (2.38)$$

to a new mixture model with fewer components. This is achieved by defining a merging function that combines different components:

$$f(\pi_i(x), \pi_j(x)) = w_{ij} \mathcal{N}(x; \mu_{ij}, P_{ij}) \quad (2.39)$$

Here,  $f$  represents the merging function for two components  $\pi_i(x)$  and  $\pi_j(x)$ , where  $w_{ij}$  is the combined weight,  $\mu_{ij}$  is the combined mean, and  $P_{ij}$  is the combined covariance. These parameters are computed as follows:

$$w_{ij} = w_i + w_j \quad (2.40)$$

$$\mu_{ij} = \frac{w_i}{w_{ij}} \mu_i + \frac{w_j}{w_{ij}} \mu_j \quad (2.41)$$

$$P_{ij} = \frac{w_i}{w_{ij}} P_i + \frac{w_j}{w_{ij}} P_j + \frac{w_i w_j}{w_{ij}^2} (\mu_i - \mu_j)(\mu_i - \mu_j)^\top \quad (2.42)$$

The GMM reduction process proceeds by iteratively merging pairs of components. To guide the merging, a bound on the KL divergence is used to quantify the information loss that occurs when two components are merged. This bound is denoted as  $B(i, j)$  and is defined for the merging of the  $i$ -th and  $j$ -th components as:

$$B(i, j) \triangleq \frac{1}{2} [w_{ij} \log |P_{ij}| - w_i \log |P_i| - w_j \log |P_j|] \quad (2.43)$$

Here,  $|\cdot|$  represents the determinant of a matrix. The primary purpose of  $B(i, j)$  is to identify the pair of components whose merging results in the smallest increase in the KL divergence value, thereby minimizing the overall information loss in the mixture. For a full derivation of this bound, refer to Section VI of [17].

It is worth noting that  $B(i, j) = B(j, i)$  and  $B(i, i) = 0$ , which simplifies the search for the optimal pair of components to merge by reducing the evaluations to  $\frac{1}{2}N(N - 1)$  combinations, where  $N$  represents the number of components in the mixture prior to reduction.

### 2.3.3.C Characteristic Function Filter

Characteristic functions (CFs) are essential tools for analyzing the behavior of random variables and their associated probability distributions. They are particularly advantageous in state estimation problems for systems influenced by non-Gaussian noise. The CF of a random variable  $\mathbf{X}$ , denoted as  $\phi_{\mathbf{X}}(\boldsymbol{\nu})$ , provides a complete characterization of its probability distribution and is defined as the expected value of  $e^{i\boldsymbol{\nu}^\top \mathbf{x}}$ , where  $i$  represents the imaginary unit,  $\boldsymbol{\nu}$  is a vector in  $\mathbb{R}^n$ , and  $f_{\mathbf{X}}(\mathbf{x})$  is the probability density function (PDF) of the random variable  $\mathbf{X}$ :

$$\mathbb{E}[e^{i\boldsymbol{\nu}^\top \mathbf{X}}] = \int_{\mathbb{R}^n} e^{i\boldsymbol{\nu}^\top \mathbf{x}} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}. \quad (2.44)$$

The CF filter operates in two main phases: the prediction step and the update step.

- **Prediction Step:** In the prediction step, given the characteristic function of the un-normalized posterior PDF after the measurement update, denoted as  $\bar{\phi}_{X_{t-1}|Z_{t-1}}(\boldsymbol{\nu})$ , the aim is to compute its propagated version,  $\bar{\phi}_{X_t|Z_{t-1}}(\boldsymbol{\nu})$ , after applying the time propagation model described by Equation 2.45:

$$x_t = Ax_{t-1} + d_t \quad (2.45)$$

$$\bar{\phi}_{X_t|Z_{t-1}}(\boldsymbol{\nu}) = \bar{\phi}_{X_{t-1}|Z_{t-1}}(A^\top \boldsymbol{\nu}) \phi_D(\boldsymbol{\nu}) \quad (2.46)$$

This equation demonstrates how the characteristic function is updated to account for the influence of the system dynamics and process noise.

- **Update Step:** For the update step, given the measurement model as described in Equation 2.47:

$$z_t = Hx_t + v_t \quad (2.47)$$

the updated characteristic function can be expressed as:

$$\bar{\phi}_{X_t|Z_t}(\boldsymbol{\nu}) = \frac{1}{(2\pi)^p} \int_{\xi} \bar{\phi}_{X_t|Z_{t-1}}(\boldsymbol{\nu} - H^T \xi) \phi_V(-\xi) e^{j\xi^T z_t} d\xi \quad (2.48)$$

This integral represents the fusion of the propagated state estimate with the new measurement, while appropriately incorporating the measurement noise through its characteristic function.

In the above equations,  $x_t \in \mathbb{R}^n$  represents the system state vector at time  $t$ , and  $z_t \in \mathbb{R}^p$  is the corresponding measurement vector. The matrices  $A \in \mathbb{R}^{n \times n}$  and  $H \in \mathbb{R}^{p \times n}$  are known system parameters that describe the state transition and observation models, respectively. The vector-valued process noise  $d_t \in \mathbb{R}^n$  is characterized by its characteristic function  $\phi_D(\boldsymbol{\nu}_d)$ , where  $\boldsymbol{\nu}_d \in \mathbb{R}^n$ . Similarly, the measurement noise  $v_t \in \mathbb{R}^p$  is described by its characteristic function  $\phi_V(\boldsymbol{\nu}_v)$ , with  $\boldsymbol{\nu}_v \in \mathbb{R}^p$ . The variable  $\xi \in \mathbb{R}^p$  serves as an integration variable in the update step.

## 2.4 Proposed Approach

This section aims to present two novel approaches to address the single sensor state estimation problem outlined in Section 2.2.

### 2.4.1 Hybrid Filter

To motivate the need for a Hybrid Filter, we begin by examining the computational challenges inherent in Bayesian filtering frameworks. As previously discussed, the Bayes Filter operates via two fundamental steps: prediction and update. Where the prediction step involves evaluating a convolution integral, which can become computationally demanding, particularly for real-time applications. One potential solution to mitigate this challenge is to employ the CF filter framework. Here, by transitioning to the frequency domain, the convolution integral in the prediction step simplifies to a multiplication operation, as per the convolution theorem.

However, while the CF framework alleviates computational difficulties in the prediction step, it introduces a convolution operation in the update step, effectively shifting the computational burden from the prediction to the update. This highlights the trade-offs associated with filtering operations in the time versus frequency domains.

To address these challenges, we propose a Hybrid Filter that capitalizes on the strengths of both the Bayesian and CF filtering frameworks. The goal is to circumvent the computational limitations of each individual approach by selectively transitioning between the time and frequency domains, while avoiding the overhead associated with repeatedly calculating Fourier transforms and their inverses at every time step. Consequently, the Hybrid Filter provides a balanced solution for SE, particularly in situations involving non-Gaussian noise.

The Hybrid Filter serves as the foundation for the two proposed approaches that will be discussed subsequently. It is therefore essential to first introduce the Hybrid Filter concept before delving into specific details of those methods. Similar to conventional filtering frameworks, the Hybrid Filter operates in two main phases: prediction and update.

To leverage the advantages of the CF Filter, the Hybrid Filter adopts the prediction step of the CF Filter. This approach is motivated by the ease of handling convolution operations in the frequency domain, where these operations become straightforward multiplications of characteristic functions. However, for the update step, the Hybrid Filter employs the update procedure from the PF. The motivation behind this choice will be clarified in subsequent sections, but it primarily relates to the PF's robustness in managing non-Gaussian noise.

The innovation of the Hybrid Filter lies in its ability to transition efficiently between the frequency and time domains. This transition step is designed to approximate the actual CF using a known CF, thereby simplifying the conversion back to the time domain. For instance, given an arbitrary CF, it can be approximated using the CF of a Gaussian distribution. Since the CF of a Gaussian has a well-known functional form, this reduces the complexity to solving an optimization problem aimed at identifying the parameters of the Gaussian distribution that best approximate the original CF.

Once these parameters are determined, both the characteristic function in the frequency domain and the corresponding PDF in the time domain can be obtained. Mathematically, this approximation can be expressed as follows:

$$\begin{aligned} \min_{\theta} \quad & \int |\phi(\boldsymbol{\nu}) - \phi_{\text{approx.}}(\boldsymbol{\nu}, \theta)|^2 d\boldsymbol{\nu} \\ \text{s.t.} \quad & \text{constraints dependent on known CF} \end{aligned} \tag{2.49}$$

where  $\phi$  and  $\phi_{\text{approx.}}$  are the actual and approximated (known) characteristic functions, respectively,  $\boldsymbol{\nu}$  is a vector in  $\mathbb{R}^n$ , and  $\theta$  is the parameter vector whose dimension depends on the specific form of the approximating CF.

This optimization problem is flexible, particularly in terms of its constraints, which depend on the nature of the known CF.

## 2.4.2 Hybrid Filter via Dirac Approximation

The Hybrid Filter via Dirac Approximation extends the Hybrid Filter concept by incorporating a discrete approximation of PDFs, derived from their corresponding CFs. Recalling Equation 2.44, the CF of a probability distribution is expressed as an expected value over kernel functions of the form  $e^{i\boldsymbol{\nu}^T \mathbf{x}}$ . This representation implies that the CF can be viewed as a weighted sum of these exponential kernel functions. Consequently, the integral defining the CF can be discretized as:

$$\phi(\boldsymbol{\nu}) \approx \sum_{k=1}^N p_k e^{i\boldsymbol{\nu}^\top \mathbf{x}_k} \quad (2.50)$$

where there are  $N$  points  $\mathbf{x}_k$ , representing discretized values of the variable  $\mathbf{X}$ , and  $p_k$  are weights corresponding to the probabilities at these points. This approximation transforms the integral into a finite sum, thereby facilitating numerical computation. Here,  $e^{i\boldsymbol{\nu}^\top \mathbf{x}_k}$  corresponds to the CF of a Dirac distribution centered at  $\mathbf{x}_k$ , effectively approximating the original PDF by a weighted sum of Dirac distributions, resulting in a discretized PDF.

To derive the discretized PDF, we establish an optimization problem that aims to align the discrete approximation of the CF with the actual CF. The objective is to minimize the squared  $L^2$ -norm of the difference between the actual CF and its approximation, formulated as follows:

$$\begin{aligned} \min_{p_k} \quad & \int \left| \phi(\boldsymbol{\nu}) - \sum_{k=1}^N p_k e^{i\boldsymbol{\nu}^\top \mathbf{x}_k} \right|^2 d\boldsymbol{\nu} \\ \text{s.t.} \quad & \sum_{k=1}^N p_k = 1 \\ & p_k \geq 0 \quad \text{for all } k = 1, \dots, N. \end{aligned} \quad (2.51)$$

This minimization problem adjusts the weights  $p_k$  to ensure that the approximation closely matches the actual CF. Constraints are imposed to guarantee that the weights  $p_k$  form a valid probability distribution.

From a computational perspective, this optimization problem can be effectively solved by discretizing the integral over a grid of points  $\boldsymbol{\nu}$ , converting the continuous integral into a summation over these grid points (Equation 2.52). This transformation reduces the problem to a constrained linear least squares problem, where the objective is to determine the weights  $p_k$  that best fit the discrete version of the CF approximation to the actual CF at these grid points. The constraints are critical to preserving the probabilistic interpretation of the PDF.

$$\begin{aligned} \min_{p_k} \quad & \sum_{l=1}^M \left| \phi(\boldsymbol{\nu}_l) - \sum_{k=1}^N p_k e^{i\boldsymbol{\nu}_l^\top \mathbf{x}_k} \right|^2 \\ \text{s.t.} \quad & \sum_{k=1}^N p_k = 1 \\ & p_k \geq 0 \quad \text{for all } k = 1, \dots, N. \end{aligned} \quad (2.52)$$

When solving this problem, careful consideration must be given to the intervals over which the discretizations are made, a challenge common to both the time and frequency domains. To address this, we first introduce the concept of moments.

In statistical analysis and probability theory, moments are critical quantitative measures that describe

the shape characteristics of a probability distribution. While the first and second moments (mean and variance, respectively) are commonly used to characterize central tendency and dispersion, higher-order moments, such as skewness and kurtosis, provide deeper insights into the asymmetry and tail behavior of the distribution. Here, we demonstrate how to compute the mean  $\boldsymbol{\mu}$  and covariance matrix  $\boldsymbol{\Sigma}$  of a random vector  $\mathbf{X}$  from the CF  $\phi_{\mathbf{X}}(\boldsymbol{\nu})$ :

$$\boldsymbol{\mu} = -i \left. \frac{d\phi_{\mathbf{X}}(\boldsymbol{\nu})}{d\boldsymbol{\nu}} \right|_{\boldsymbol{\nu}=0} \quad (2.53)$$

$$\boldsymbol{\Sigma} = - \left. \frac{d^2\phi_{\mathbf{X}}(\boldsymbol{\nu})}{d\boldsymbol{\nu}^2} \right|_{\boldsymbol{\nu}=0} - \boldsymbol{\mu}\boldsymbol{\mu}^T \quad (2.54)$$

To determine the interval of discretization over  $\mathbf{X}$ , we utilize the moment information contained in the original CF. Specifically, we extract the first and second moments using Equations 2.53 and 2.54, respectively. Then, we calculate the interval  $I_j$  for each dimension  $j$  of the grid as follows:

$$I_j = \left[ \mu_j - 3.5\Sigma_{j,j}^{\frac{1}{2}}, \mu_j + 3.5\Sigma_{j,j}^{\frac{1}{2}} \right]. \quad (2.55)$$

The discretization intervals over  $\boldsymbol{\nu}$  are determined using an approximation based on the CF of a zero-mean Gaussian distribution. Given that the significant portion of the CF is within the range where its magnitude is non-negligible, we set a threshold  $\epsilon$  to define this effective range. Assuming a zero-mean Gaussian CF, the magnitude decays as  $e^{-\frac{\boldsymbol{\nu}^T \boldsymbol{\Sigma} \boldsymbol{\nu}}{2}}$ , where  $\boldsymbol{\Sigma}$  is the covariance matrix. The threshold is set where the magnitude of the CF approximates zero:

$$e^{-\frac{\boldsymbol{\nu}^T \boldsymbol{\Sigma} \boldsymbol{\nu}}{2}} \leq \epsilon \quad (2.56)$$

Taking the logarithm of both sides and solving for  $\boldsymbol{\nu}$ , we obtain:

$$\boldsymbol{\nu}^T \boldsymbol{\Sigma} \boldsymbol{\nu} \geq 2 \log \left( \frac{1}{\epsilon} \right) \quad (2.57)$$

For each dimension  $j$ , we approximate the range of  $\nu_j$  by considering only the diagonal elements  $\Sigma_{j,j}$ , simplifying the expression to:

$$|\nu_j| \geq \sqrt{\frac{2 \log \left( \frac{1}{\epsilon} \right)}{\Sigma_{j,j}}} \quad (2.58)$$

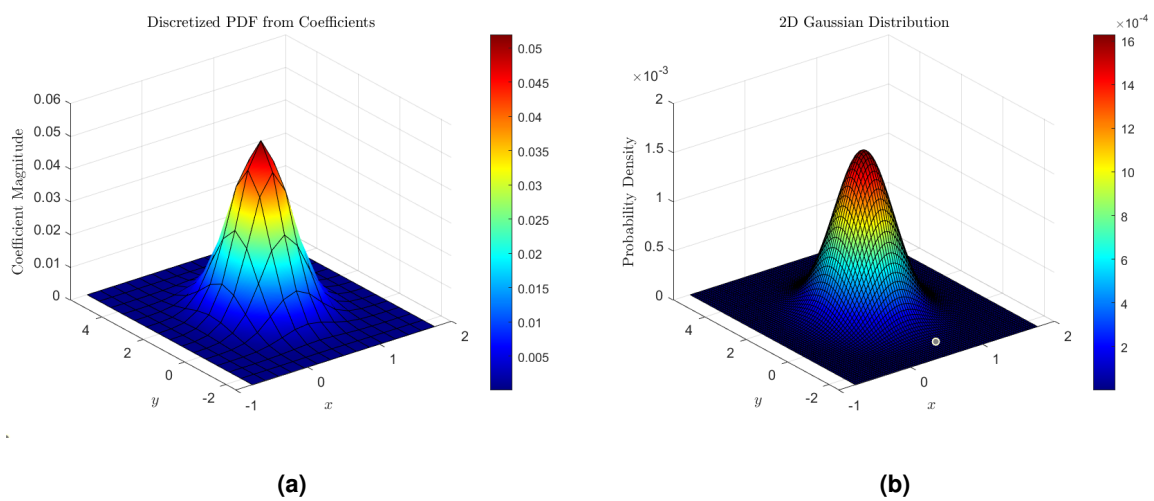
This approximation provides a practical means of selecting discretization intervals over  $\boldsymbol{\nu}$  that are expected to capture the essential behavior of the CF for the purpose of estimating the PDF. Once these intervals are determined, uniform random sampling is employed to generate the  $N$  points  $\mathbf{x}_k$  and the  $M$  points  $\boldsymbol{\nu}_l$ .

Following this approximation, we obtain a discretized representation of the PDF, which corresponds

to particles within the PF framework. This is where the PF update step becomes particularly useful, as it can now operate on these particles. After completing this step, it is possible to transition back to the frequency domain by incorporating the updated particles and their associated weights into the CF expression, as shown in Equation 2.50, substituting the  $x_k$  and  $p_k$  values accordingly. Thus, a full iteration of the Hybrid Filter via Dirac Approximation is completed.

### 2.4.2.A Examples

Some examples of the results obtained for the approximations of different PDFs by their CFs, using the method described in the previous Sub-section, are presented below.



**Figure 2.1:** Gaussian *PDF*, approximation on the left and real *PDF* on the right.

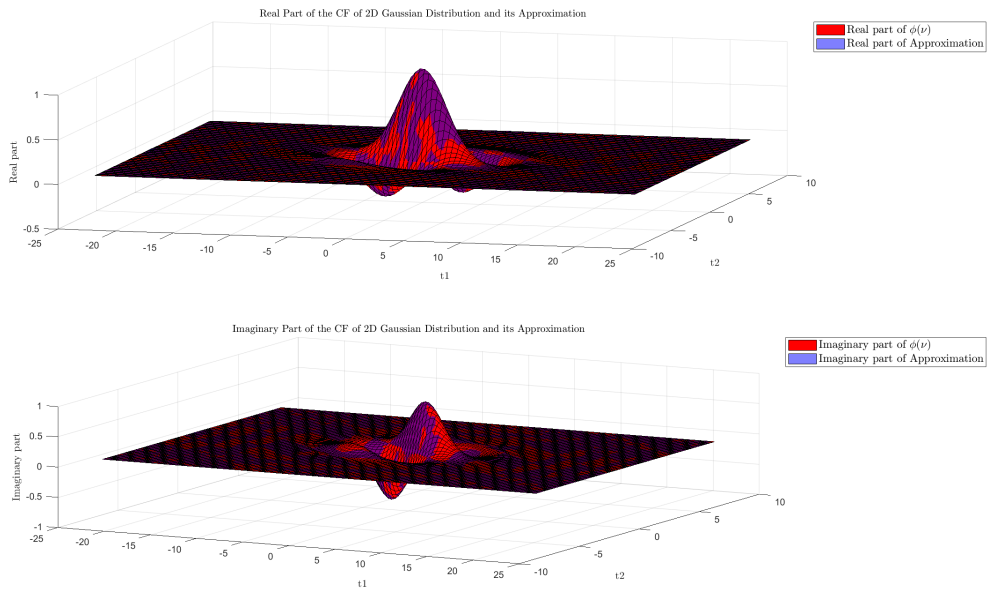


Figure 2.2: Real and approximated CF's of gaussian random variable.

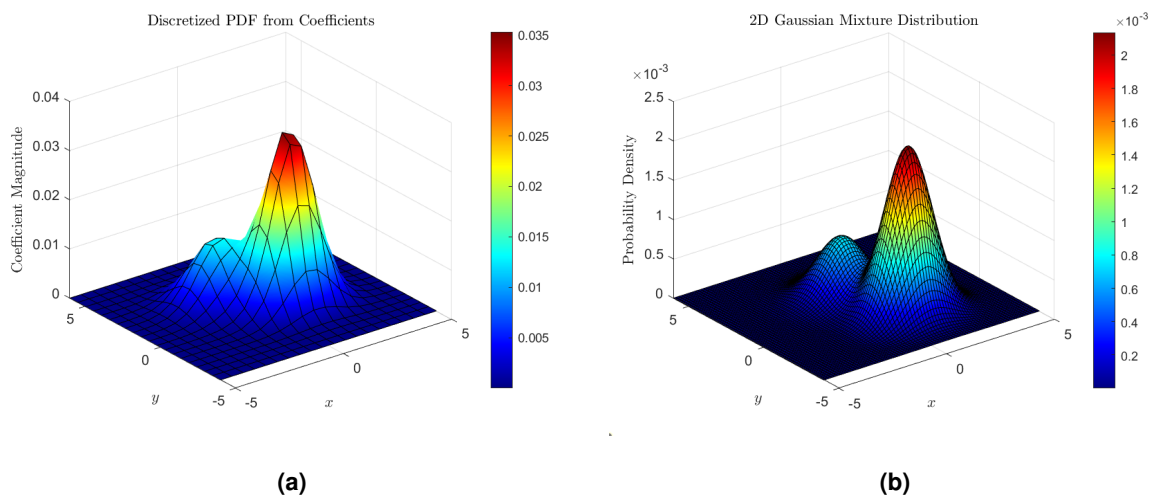


Figure 2.3: Gaussian Mixture PDF, approximation on the left and real PDF on the right.

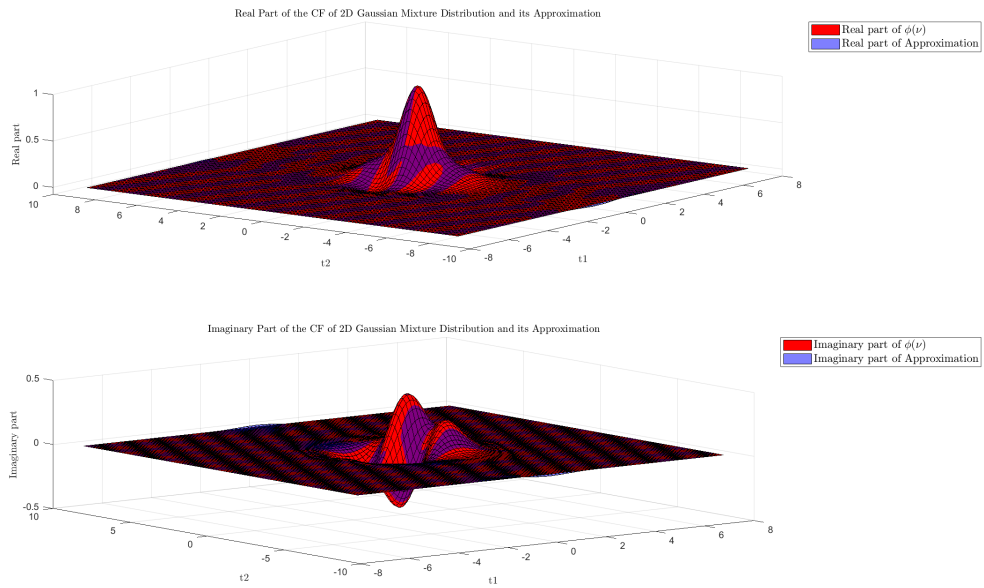


Figure 2.4: Real and approximated *CF*'s of gaussian mixture random variable.

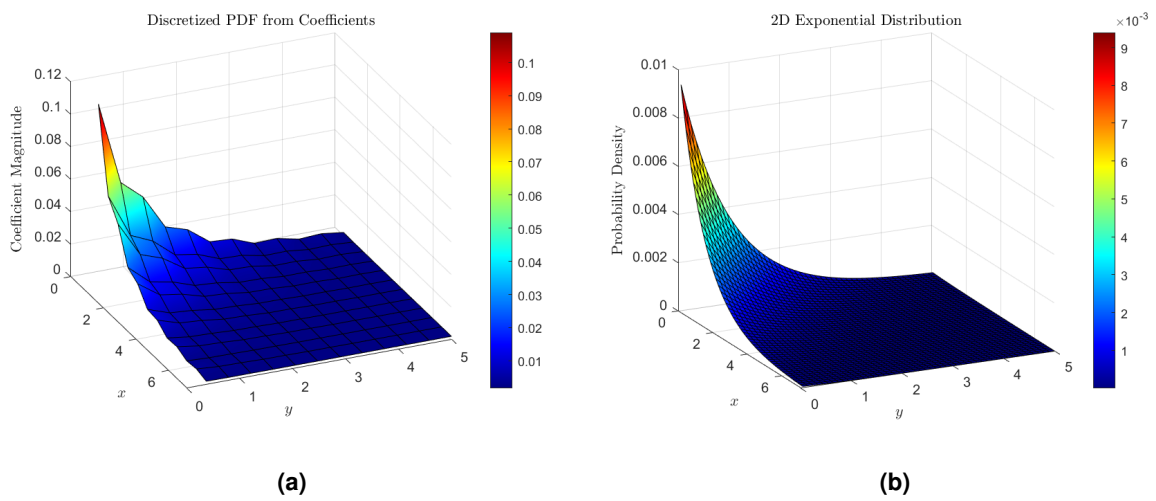


Figure 2.5: Exponential *PDF*, approximation on the left and real *PDF* on the right.

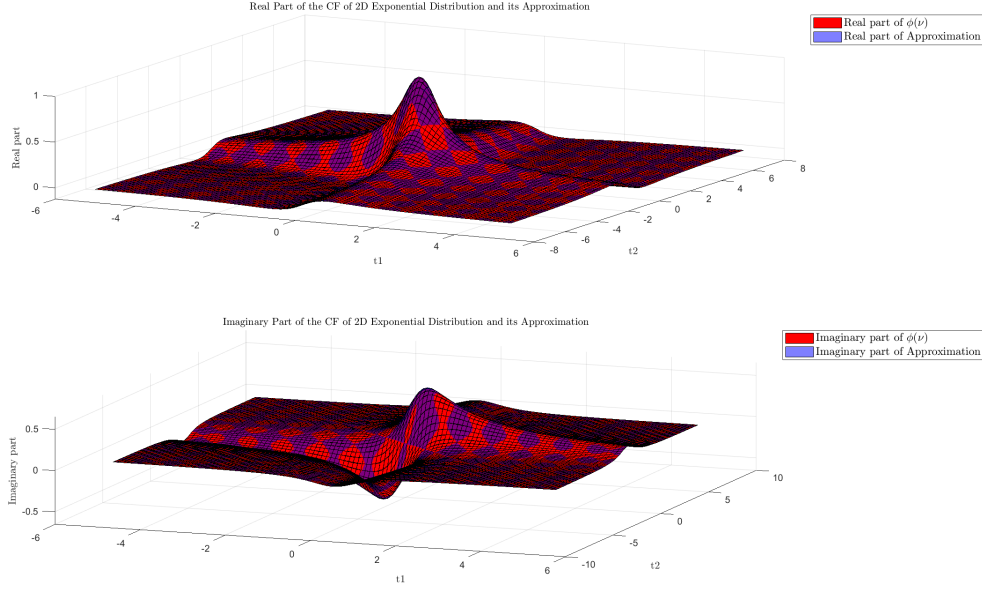


Figure 2.6: Real and approximated  $CF$ 's of exponential random variable.

### 2.4.3 Hybrid Filter via Gaussian Mixture Approximation

Building on the foundational concept of the Hybrid Filter, this method introduces an approximation utilizing Gaussian kernels rather than Dirac kernels. The resulting approach approximates the PDF through a mixture of Gaussian components, which corresponds to the characteristic function (CF) of a Gaussian Mixture Model (GMM).

The CF of a Gaussian distribution centered at  $\boldsymbol{\mu}$  with covariance  $\boldsymbol{\Sigma}$  is given by:

$$\phi_{\text{Gaussian}}(\boldsymbol{\nu}) = e^{i\boldsymbol{\nu}^\top \boldsymbol{\mu}} e^{-\frac{1}{2}\boldsymbol{\nu}^\top \boldsymbol{\Sigma} \boldsymbol{\nu}}. \quad (2.59)$$

Thus, the CF of a GMM can be expressed as a weighted sum of the individual CFs of its Gaussian components. Given that any arbitrary distribution can be effectively approximated by a GMM with a sufficient number of Gaussian components, the approximation of the original CF can be represented as follows:

$$\phi(\boldsymbol{\nu}) \approx \sum_{k=1}^N p_k e^{i\boldsymbol{\nu}^\top \boldsymbol{\mu}_k} e^{-\frac{1}{2}\boldsymbol{\nu}^\top \boldsymbol{\Sigma}_k \boldsymbol{\nu}}, \quad (2.60)$$

where  $N$  is the number of Gaussian components,  $\boldsymbol{\mu}_k$  and  $\boldsymbol{\Sigma}_k$  are the mean vector and covariance matrix of the  $k$ -th Gaussian, and  $p_k$  are the corresponding weights.

To approximate the PDF, we formulate a minimization problem akin to the Dirac approximation ap-

proach. Here, the objective is to minimize the squared  $L^2$ -norm of the difference between the actual CF and its Gaussian mixture approximation:

$$\begin{aligned}
& \min_{p_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k} \int \left| \phi(\boldsymbol{\nu}) - \sum_{k=1}^N p_k e^{i\boldsymbol{\nu}^\top \boldsymbol{\mu}_k} e^{-\frac{1}{2}\boldsymbol{\nu}^\top \boldsymbol{\Sigma}_k \boldsymbol{\nu}} \right|^2 d\boldsymbol{\nu} \\
& \text{s.t.} \quad \sum_{k=1}^N p_k = 1 \\
& \quad p_k \geq 0 \quad \text{for all } k = 1, \dots, N, \\
& \quad \boldsymbol{\Sigma}_k \succ 0 \quad \text{for all } k = 1, \dots, N.
\end{aligned} \tag{2.61}$$

The constraints ensure that the weights  $p_k$  form a valid GMM and that the covariance matrices  $\boldsymbol{\Sigma}_k$  are positive definite. This optimization problem is inherently more challenging than the Dirac-based approach, as it requires solving for not only the weights  $p_k$ , but also the parameters  $\boldsymbol{\mu}_k$  and  $\boldsymbol{\Sigma}_k$  for each Gaussian component.

In practice, solving this optimization problem involves discretizing the domain  $\boldsymbol{\nu}$  into grid points and transforming the continuous integral into a finite summation over these points (see Equation 2.62). This transformation reduces the problem to a constrained non-linear optimization, which can be addressed using standard numerical methods, such as gradient-based approaches.

$$\begin{aligned}
& \min_{p_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k} \sum_{l=1}^M \left| \phi(\boldsymbol{\nu}_l) - \sum_{k=1}^N p_k e^{i\boldsymbol{\nu}_l^\top \boldsymbol{\mu}_k} e^{-\frac{1}{2}\boldsymbol{\nu}_l^\top \boldsymbol{\Sigma}_k \boldsymbol{\nu}_l} \right|^2 \\
& \text{s.t.} \quad \sum_{k=1}^N p_k = 1 \\
& \quad p_k \geq 0 \quad \text{for all } k = 1, \dots, N, \\
& \quad \boldsymbol{\Sigma}_k \succ 0 \quad \text{for all } k = 1, \dots, N.
\end{aligned} \tag{2.62}$$

Furthermore, we can leverage the moments of the CF to guide the initialization of the Gaussian parameters. The mean vector  $\boldsymbol{\mu}$  and covariance matrix  $\boldsymbol{\Sigma}$  of the underlying distribution can be directly derived from the CF using Equations 2.53 and 2.54.

Similar to the Dirac-based approach, these moments can be used to define the discretization intervals (Equations 2.55 and 2.58), providing the ranges needed to generate the  $N$  points  $\mathbf{x}_k$  and the  $M$  points  $\boldsymbol{\nu}_l$  via uniform random sampling.

Once the Gaussian mixture approximation is established, we proceed by sampling from it to obtain a discretized representation of the PDF. This discretized representation is then utilized in the PF update step. Following the update, we transition back to the frequency domain using the same procedure as in the Dirac-based approach, thereby completing a full iteration of the Hybrid Filter via Gaussian Mixture Approximation.

## 2.5 Experimental Results and Discussion

This section presents the results of simulations conducted to evaluate the single sensor problem, focusing on the application of the Hybrid Filter. The simulations are detailed in Subsection 2.5.1, followed by a discussion of the results in Subsection 2.5.2.

### 2.5.1 Simulation

This subsection outlines the simulations conducted to evaluate the performance of the proposed Hybrid Filter in the single-sensor problem context. The primary objective is to estimate the position of a vehicle undergoing a random walk, modeled using discrete double integrator dynamics with a sampling time of  $T_s = 0.1$  s. These simulations are designed to assess the filter's effectiveness under various assumptions regarding the input and observation noise distributions.

In all simulations, the state-space model used aligns with that defined in the problem formulation (Equations 2.1 and 2.2), with one notable difference: no external disturbance  $d_t$  is considered. Instead, at each time step, the control input  $u_t$  is sampled from either a Gaussian distribution or a GMM with two or three components. The observation model remains consistent across all simulations, with the observation matrix  $H$  defined as:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The observation noise  $v_t$  is sampled from one of three distributions: a Gaussian distribution, a Gaussian mixture, or an exponential distribution. These variations in both process and observation noise provide a comprehensive testbed for evaluating the Hybrid Filter under diverse noise scenarios.

The aim of each simulation is to estimate the position of the vehicle as it undergoes a random walk. Since the initial state of the vehicle is not precisely known, it is modeled as a zero-mean Gaussian distribution. Both the Dirac-based and Gaussian Mixture approaches of the Hybrid Filter were tested across all combinations of random walks (Gaussian or GMM) and observation noise types (Gaussian, GMM, and exponential). Consequently, nine distinct scenarios were evaluated:

- Gaussian walk with Gaussian noise.
- Gaussian walk with exponential noise.
- Gaussian walk with GMM noise.
- GMM (2 peaks) walk with Gaussian noise.
- GMM (2 peaks) walk with exponential noise.
- GMM (2 peaks) walk with GMM noise.
- GMM (3 peaks) walk with GMM noise.
- GMM (3 peaks) walk with Gaussian noise.
- GMM (3 peaks) walk with exponential noise.
- GMM (3 peaks) walk with GMM noise.

For each of these scenarios, the Dirac approach was tested with different numbers of particles, while the Gaussian Mixture approach varied the number of Gaussian components in the mixture, keeping the particle count fixed at 256.

To provide a benchmark, both the KF and PF were used as comparative methods. The KF represents an optimal solution under the Gaussian assumption, while the PF was configured with the same number of particles as the Hybrid Filter to ensure a fair comparison.

The filters were evaluated over 50 independent runs, each consisting of 150 time steps. In each run, the vehicle's initial velocity was set to  $[0 \ 0]^T$ , while its initial position was uniformly sampled within the square defined by vertices at  $(-8, -8)$ ,  $(-8, -10)$ ,  $(-10, -10)$ , and  $(-10, -8)$ . During each run, the type of random walk and observation noise remained fixed, allowing for a controlled assessment of filter performance under specific conditions.

The performance of each filter was evaluated using two key metrics: cumulative Root Mean Square Error (RMSE) over the 150 time steps to assess estimation accuracy, and the trace of the covariance matrix to evaluate the uncertainty associated with the estimates at each time step.

These simulations provide a rigorous framework for assessing the Hybrid Filter's capabilities under varying noise conditions. By comparing the performance of the Hybrid Filter with that of the KF and PF, the robustness of the proposed approach in both Gaussian and non-Gaussian noise environments is demonstrated. The results, presented in the following subsection, provide insights into the effectiveness of both the Dirac and Gaussian Mixture approaches.

## 2.5.2 Results

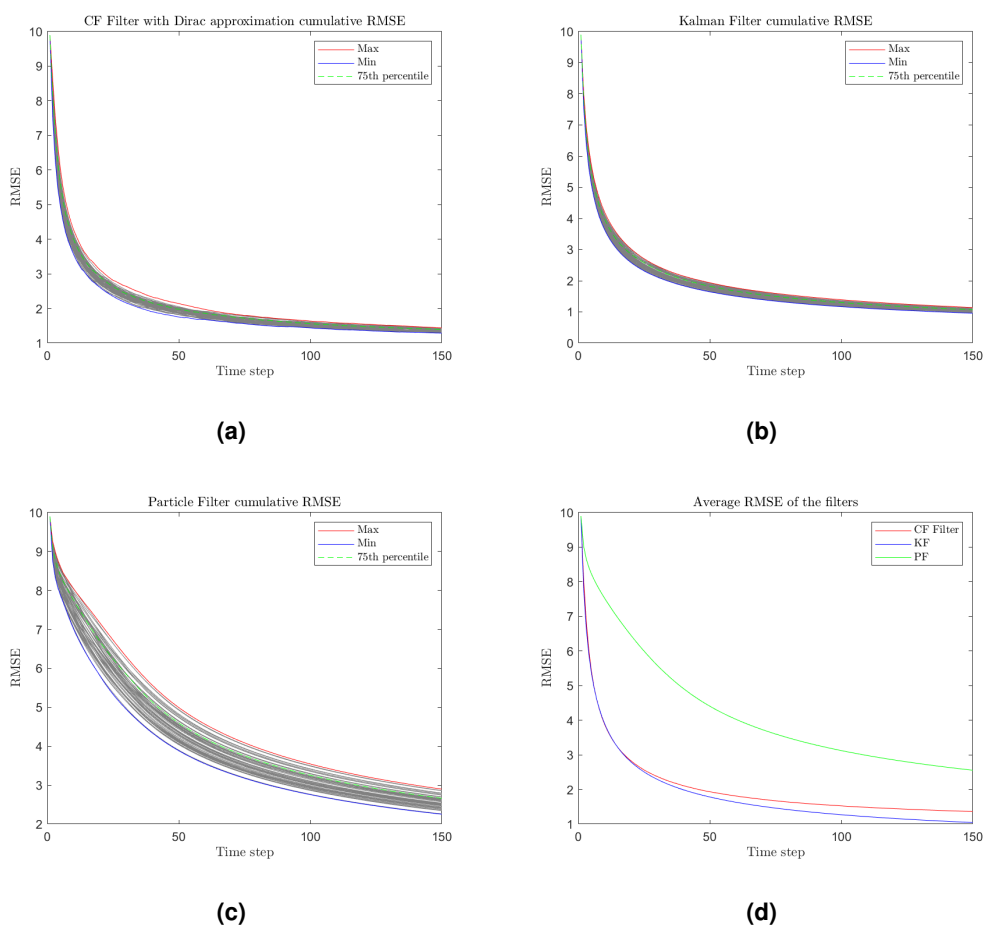
In this subsection, we present and discuss the results obtained for the Hybrid Filter using both the Dirac and Gaussian Mixture approaches. For simplicity, we refer to the Hybrid Filter as the CF Filter.

### 2.5.2.A Hybrid Filter via Dirac Approximation

For the Dirac approximation approach, the nine distinct scenarios were tested with different numbers of particles—specifically, 256, 625, and 1296—resulting in a total of 27 simulations. Here, we present and discuss only the most pertinent results.

The performance of the CF Filter in the Gaussian walk with Gaussian noise scenario is illustrated in Figure 2.7. The first three plots show the cumulative RMSE of each filter across all 50 runs (represented by gray lines). These plots also indicate the upper and lower performance bounds for each filter, denoted by the maximum and minimum lines. The 75<sup>th</sup> percentile of the RMSE is represented in green to provide further insight into the overall performance. Finally, for ease of comparison, Figure 2.7(d) shows the average cumulative RMSE at each time step, averaged over the 50 runs.

The CF Filter demonstrates strong performance, closely following the KF, which is expected to yield optimal results in this linear Gaussian setting. This outcome is particularly noteworthy as it highlights the capability of the CF Filter to remain competitive even in scenarios where the KF benefits from assumptions that perfectly align with its design. Given that the CF Filter is also capable of handling non-Gaussian noise distributions, its close performance to the KF in this Gaussian scenario suggests robustness and adaptability across a wide range of noise types. Moreover, the CF Filter exhibits consistent behavior across the 50 runs, as evidenced by the tight clustering of error bounds in Figure 2.7(a). This consistency implies that the CF Filter maintains reliability and robustness even under varying conditions.



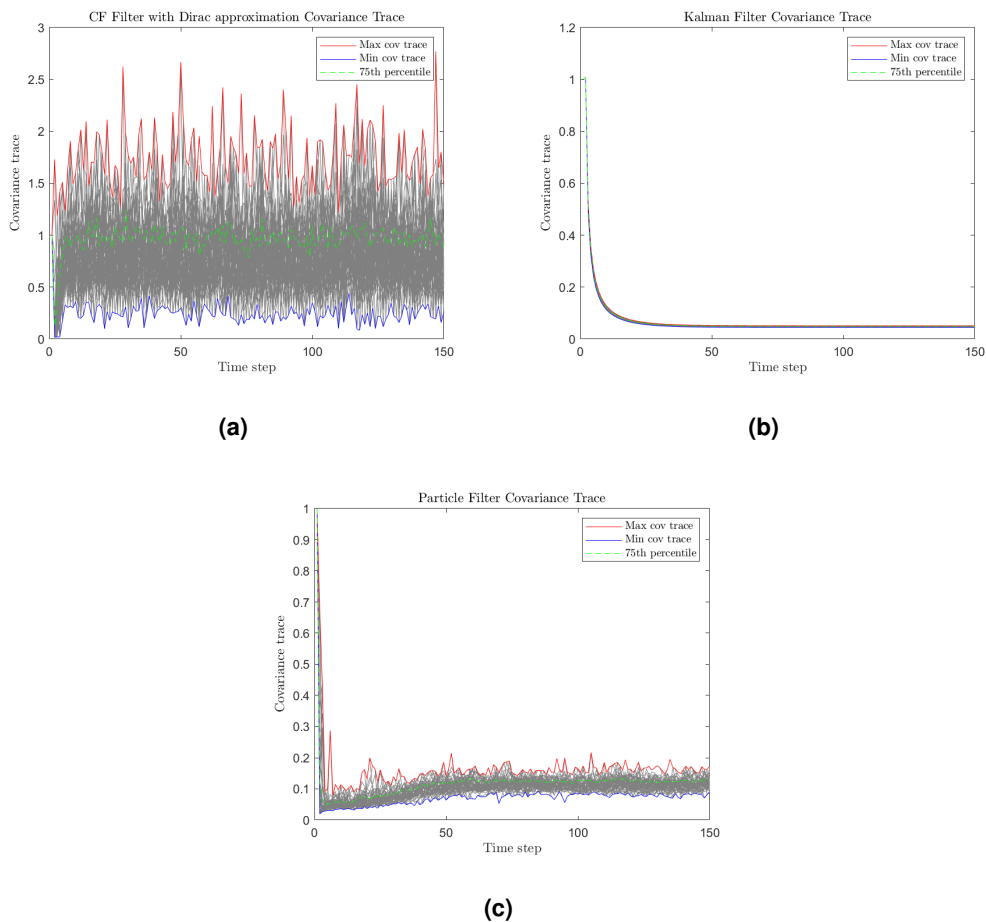
**Figure 2.7:** Cumulative RMSE of the position of Gaussian walk with Gaussian noise: a) CF Filter with 256 particles; b) Kalman Filter; c) Particle Filter with 256 particles; d) Average of the cumulative RMSE over the 50 runs.

In contrast, the PF demonstrates relatively weaker performance. While the PF theoretically has the capability to manage non-Gaussian noise and nonlinearities, its effectiveness in this scenario is limited by the relatively small particle count (256). This constraint likely results in higher variance in the estimation results and a greater overall RMSE, as evidenced in Figure 2.7(c). The reliance of the

PF on a sufficiently large number of particles for accurate state estimation thus becomes a significant bottleneck, especially when compared to the CF Filter, which demonstrates superior performance under the same conditions.

Figure 2.7(d) provides a direct comparison of the average cumulative RMSE for each filter across 50 runs. While the KF yields the best performance, the CF Filter achieves results that are nearly indistinguishable from those of the KF, further validating its applicability as a viable alternative, particularly in environments where the Gaussian noise assumption might not hold. The PF, on the other hand, falls significantly behind, underscoring the limitations of particle-based methods when computational resources are constrained.

The covariance trace results in Figures 2.8(a), 2.8(b), and 2.8(c) illustrate key differences in how each filter manages estimation uncertainty.



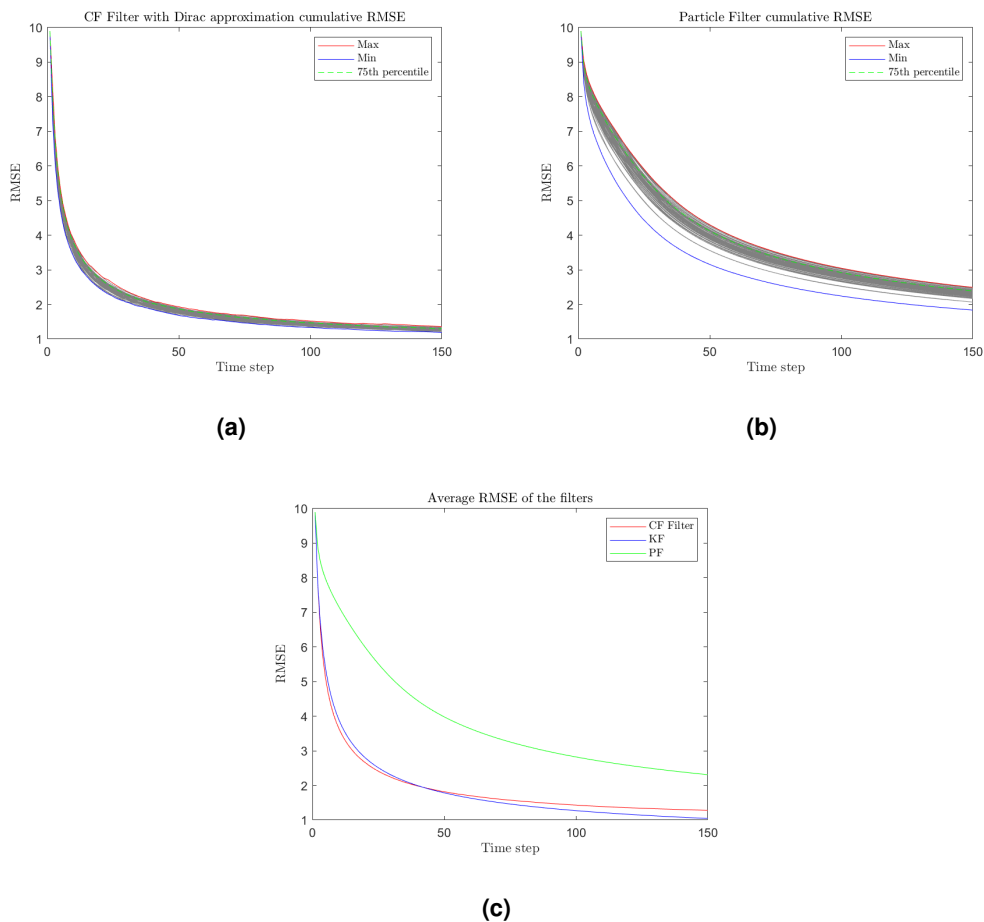
**Figure 2.8:** Covariance Matrix trace of the position estimate uncertainty in the Gaussian walk with Gaussian noise: a) CF Filter with 256 particles; b) Kalman Filter; c) Particle Filter with 256 particles.

The CF Filter, while effective in state estimation, does not prioritize minimizing estimation uncertainty

to the same degree as the KF. The KF, designed specifically for linear systems with Gaussian noise, optimizes its approach to reduce uncertainty rapidly, leading to consistently low covariance trace values. In contrast, the broader design focus of the CF Filter results in a less aggressive reduction in uncertainty, which translates to greater variability across runs.

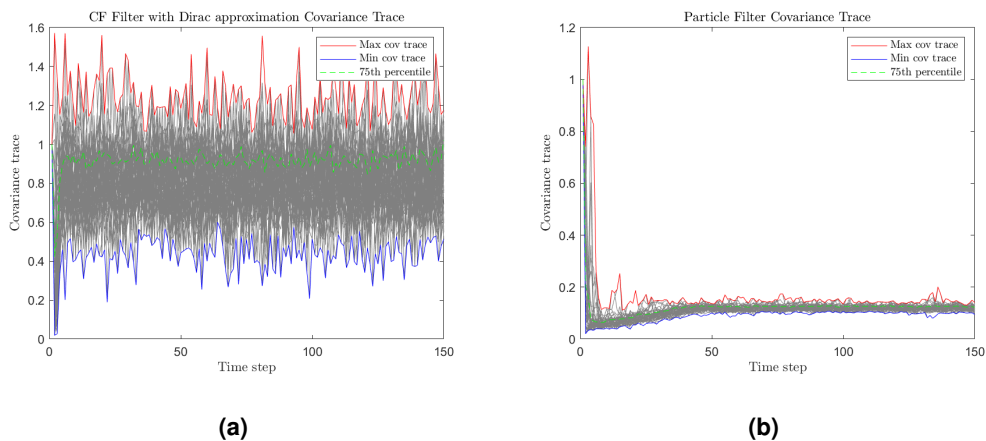
Although the PF displays greater variability compared to the KF, its overall uncertainty remains relatively low. This can be attributed to the inherent nature of particle filters, which represent state distributions by sampling a large number of particles. However, the limited particle count introduces fluctuations in the covariance trace, indicating the need for more particles to achieve smoother, more reliable performance.

To evaluate the impact of increasing the particle count, we turn to Figures 2.9 and 2.10. These figures demonstrate improved performance for both the CF Filter and the PF. Notably, the cumulative RMSE is reduced, and the error bounds in Figures 2.9(a) and 2.9(b) are narrower, indicating increased consistency across runs.



**Figure 2.9:** Cumulative RMSE of the position of Gaussian walk with Gaussian noise: a) CF Filter with 1296 particles; b) Particle Filter with 1296 particles; c) Average of the cumulative RMSE over the 50 runs.

A similar trend is observed in Figures 2.10(a) and 2.10(b), where the average covariance trace decreases slightly and the uncertainty bounds become tighter, further highlighting the benefits of increasing the particle count.



**Figure 2.10:** Covariance Matrix trace of the position estimate uncertainty in the Gaussian walk with Gaussian noise: a) CF Filter with 1296 particles; b) Particle Filter with 1296 particles.

In the scenario where the CF Filter is tested with a Gaussian mixture walk and exponential noise using 1296 particles, the results indicate a marked shift in performance compared to the Gaussian noise setting. The cumulative RMSE results, presented in Figures 2.11(a), 2.11(b), and 2.11(c), offer a comprehensive view of each filter’s behavior over 50 runs.

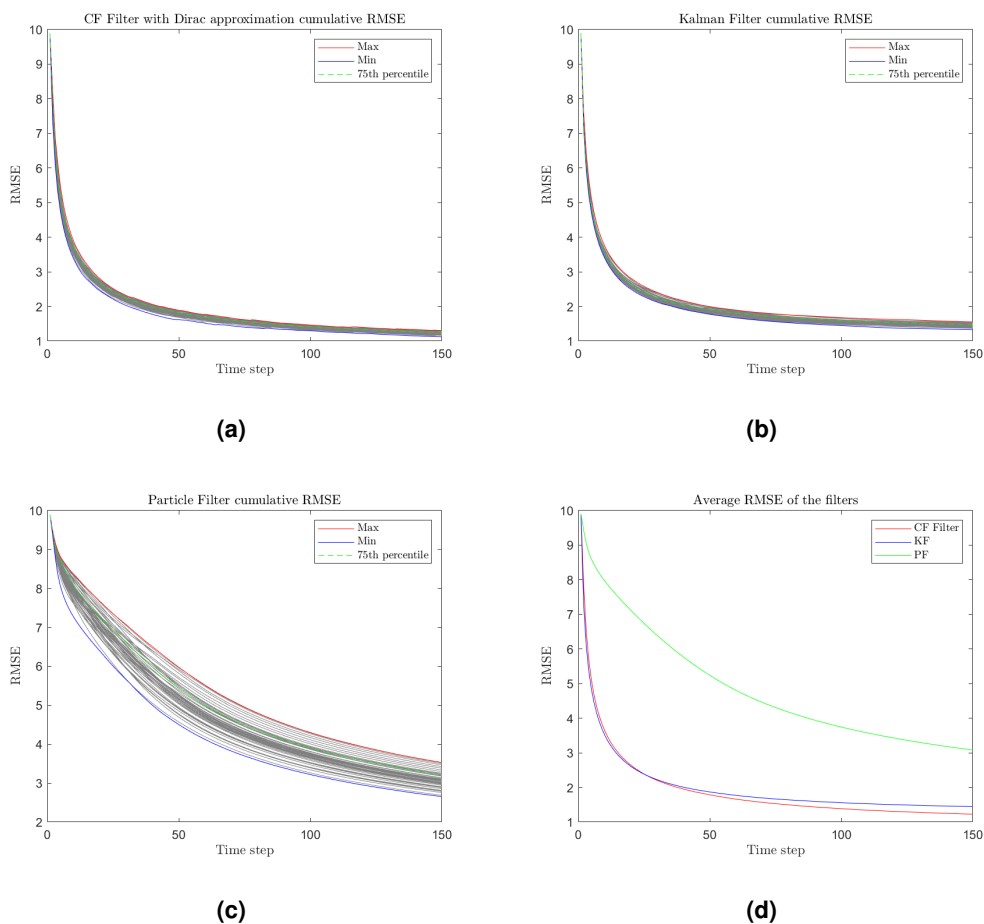
The CF Filter exhibits strong performance, now surpassing the KF, which is no longer in its optimal operating environment due to the introduction of non-Gaussian, exponential noise. The inherent adaptability of the CF Filter becomes more evident in this scenario, where its capability to handle non-Gaussian noise allows it to outperform the KF. The tighter error bounds and reduced variability across runs (as observed in Figure 2.11(a)) underscore the robustness of the CF Filter, which manages to maintain consistent accuracy across a more complex noise distribution.

In comparison, the KF, which is designed primarily for Gaussian noise environments, exhibits a decline in relative performance. This outcome is consistent with theoretical expectations, as the KF struggles under non-Gaussian conditions where its underlying noise assumptions are no longer valid.

The PF, despite being theoretically suitable for handling non-Gaussian noise, continues to show relatively weaker performance. This can be attributed to the number of particles (1296), which still seems insufficient for accurately capturing the underlying noise characteristics and system dynamics. The higher variance and broader error bounds (as depicted in Figure 2.11(c)) suggest that the PF struggles to consistently match the performance of the CF Filter.

The overall comparison of the average cumulative RMSE in Figure 2.11(d) reinforces the superiority

of the CF Filter in this context. Its ability to outperform both the KF and the PF underscores the strength of its Dirac approximation, enabling efficient handling of non-Gaussian noise distributions. While the KF remains competitive under simpler noise structures, it is less effective in this more complex setting, and the PF remains constrained by the limitations imposed by particle count.



**Figure 2.11:** Cumulative RMSE of the position of Gaussian Mixture walk with Exponential noise: a) CF Filter with 1296 particles; b) Kalman Filter; c) Particle Filter with 1296 particles; d) Average of the cumulative RMSE over the 50 runs.

To finalize the analysis of the results from the Dirac approach, we refer to Table 2.1, which provides a summary of the performance of each filter across different scenarios. The table reports both the mean RMSE and its standard deviation over the 50 runs, with the best results for each scenario highlighted in bold.

From the table, it is evident that the CF Filter consistently achieves the best performance in scenarios involving exponential noise. This is further underscored by its low standard deviation, suggesting that the filter maintains robust performance across different runs. Such consistency can be attributed to the inherent flexibility of the CF Filter, which is well-adapted to handle non-Gaussian noise types.

Additionally, the results illustrate the beneficial impact of increasing the particle count on the CF Filter's performance. As the number of particles grows, the RMSE decreases, indicating an improved capacity of the filter to approximate the underlying system dynamics more accurately.

One notable observation from the table is that, in Gaussian mixture noise scenarios, the KF continues to outperform the CF Filter. This outcome can be explained by the fact that the Gaussian mixture noise used in these simulations is not highly complex and can be effectively approximated by a zero-mean Gaussian distribution. As such, the KF performs optimally under these conditions, leveraging its inherent strengths in Gaussian noise environments. While the performance gap is not particularly large, it nonetheless underscores the KF's advantage when operating under conditions that closely align with its assumptions.

The PF, on the other hand, continues to exhibit weaker performance across most scenarios. Although the PF is theoretically well-suited to handle non-Gaussian noise and complex system dynamics, its effectiveness here is constrained by the relatively low particle count. Even with an increased number of particles, the PF fails to match the performance of either the CF Filter or the KF, suggesting that it struggles to capture the full spectrum of system behaviors under these conditions. The higher variance in RMSE across the runs further highlights the limitations of the PF, as it requires a significantly larger particle count to achieve a more consistent and accurate estimate.

|                  |   | Simulation Scenario              | CF Filter w/ 256 particles | CF Filter w/ 625 particles | CF Filter w/ 1296 particles | KF            | PF w/ 256 particles | PF w/ 625 particles | PF w/ 1296 particles |
|------------------|---|----------------------------------|----------------------------|----------------------------|-----------------------------|---------------|---------------------|---------------------|----------------------|
| <b>Mean RMSE</b> | <b>Average over 50 simulations</b>            | G. Walk w/ G. Noise              | 1.3635                     | 1.3155                     | 1.2808                      | <b>1.0579</b> | 2.5559              | 2.4408              | 2.3117               |
|                  |   | G. Walk w/ GM Noise              | 1.3643                     | 1.3262                     | 1.2915                      | <b>1.0871</b> | 2.5236              | 2.4160              | 2.2951               |
|                  |   | G. Walk w/ Exp. Noise            | 1.2754                     | 1.2395                     | <b>1.2263</b>               | 1.4449        | 3.4771              | 3.2391              | 3.0742               |
|                  |   | GM w/ 2 peaks Walk w/ G. Noise   | 1.3635                     | 1.3081                     | 1.2729                      | <b>1.0410</b> | 2.5611              | 2.4158              | 2.2951               |
|                  |   | GM w/ 2 peaks Walk w/ GM Noise   | 1.3712                     | 1.3182                     | 1.2789                      | <b>1.0940</b> | 2.5742              | 2.3881              | 2.3176               |
|                  |   | GM w/ 2 peaks Walk w/ Exp. Noise | 1.2917                     | 1.2400                     | <b>1.2210</b>               | 1.3831        | 3.4365              | 3.1862              | 3.0585               |
|                  |   | GM w/ 3 peaks Walk w/ G. Noise   | 1.3626                     | 1.3166                     | 1.2885                      | <b>1.0488</b> | 2.5717              | 2.3875              | 2.2942               |
|                  |   | GM w/ 3 peaks Walk w/ GM Noise   | 1.3763                     | 1.3213                     | 1.2875                      | <b>1.0934</b> | 2.5918              | 2.4245              | 2.3195               |
|                  |   | GM w/ 3 peaks Walk w/ Exp. Noise | 1.2868                     | 1.2464                     | <b>1.2284</b>               | 1.4446        | 3.4472              | 3.2467              | 3.0810               |
|                  | <b>Standard deviation over 50 simulations</b> | G. Walk w/ G. Noise              | 0.0400                     | 0.0348                     | <b>0.0344</b>               | 0.0418        | 0.1572              | 0.1125              | 0.1225               |
|                  |   | G. Walk w/ GM Noise              | 0.0360                     | 0.0404                     | 0.0373                      | <b>0.0358</b> | 0.1555              | 0.1487              | 0.1312               |
|                  |   | G. Walk w/ Exp. Noise            | 0.0471                     | <b>0.0352</b>              | 0.0362                      | 0.0454        | 0.2897              | 0.2810              | 0.2048               |
|                  |   | GM w/ 2 peaks Walk w/ G. Noise   | 0.0462                     | 0.0376                     | <b>0.0356</b>               | 0.0361        | 0.1927              | 0.1397              | 0.1238               |
|                  |   | GM w/ 2 peaks Walk w/ GM Noise   | 0.0405                     | 0.0408                     | <b>0.0311</b>               | 0.0422        | 0.1248              | 0.1433              | 0.0935               |
|                  |   | GM w/ 2 peaks Walk w/ Exp. Noise | 0.0417                     | 0.0420                     | <b>0.0395</b>               | 0.0474        | 0.2594              | 0.2451              | 0.1845               |
|                  |   | GM w/ 3 peaks Walk w/ G. Noise   | 0.0451                     | 0.0389                     | <b>0.0296</b>               | 0.0380        | 0.1490              | 0.1338              | 0.1143               |
|                  |   | GM w/ 3 peaks Walk w/ GM Noise   | 0.0337                     | 0.0328                     | <b>0.0298</b>               | 0.0485        | 0.1575              | 0.1295              | 0.1108               |
|                  |   | GM w/ 3 peaks Walk w/ Exp. Noise | 0.0470                     | <b>0.0350</b>              | 0.0394                      | 0.0400        | 0.2687              | 0.2353              | 0.2083               |

**Table 2.1:** RMSE of all filters along different simulation scenarios, Dirac approach.

### 2.5.2.B Hybrid Filter via Gaussian Mixture Approximation

For the Gaussian mixture approach, nine distinct scenarios were tested with the number of Gaussians used in the approximation varying between three and four, resulting in a total of 18 simulations. Here, we present and discuss only the most pertinent results.

The results of the CF Filter via the Gaussian mixture approximation with three Gaussians, tested in a scenario involving both a Gaussian mixture walk and Gaussian mixture noise, are illustrated in Figures 2.12(a), 2.12(b), 2.12(c), and 2.12(d). The focus here is on assessing how the CF Filter, based on the Gaussian mixture approximation, performs relative to the KF and PF across these 50 simulation runs.

As seen in Figure 2.12(a), the CF Filter employing the Gaussian mixture approximation with three Gaussians demonstrates a more erratic performance compared to the results obtained using the Dirac approach. The cumulative RMSE reveals fluctuations across different runs, as indicated by the larger spread between maximum and minimum error bounds. These variations in performance can be attributed to the inherent challenges of the Gaussian mixture approximation itself. Specifically, the non-convex nature of the optimization problem involved in determining the Gaussian mixture may lead to convergence to local minima, thereby preventing the optimal representation of the system state. Additionally, it may reflect that the selected number of Gaussian components is insufficient to accurately capture the complexity of the state.

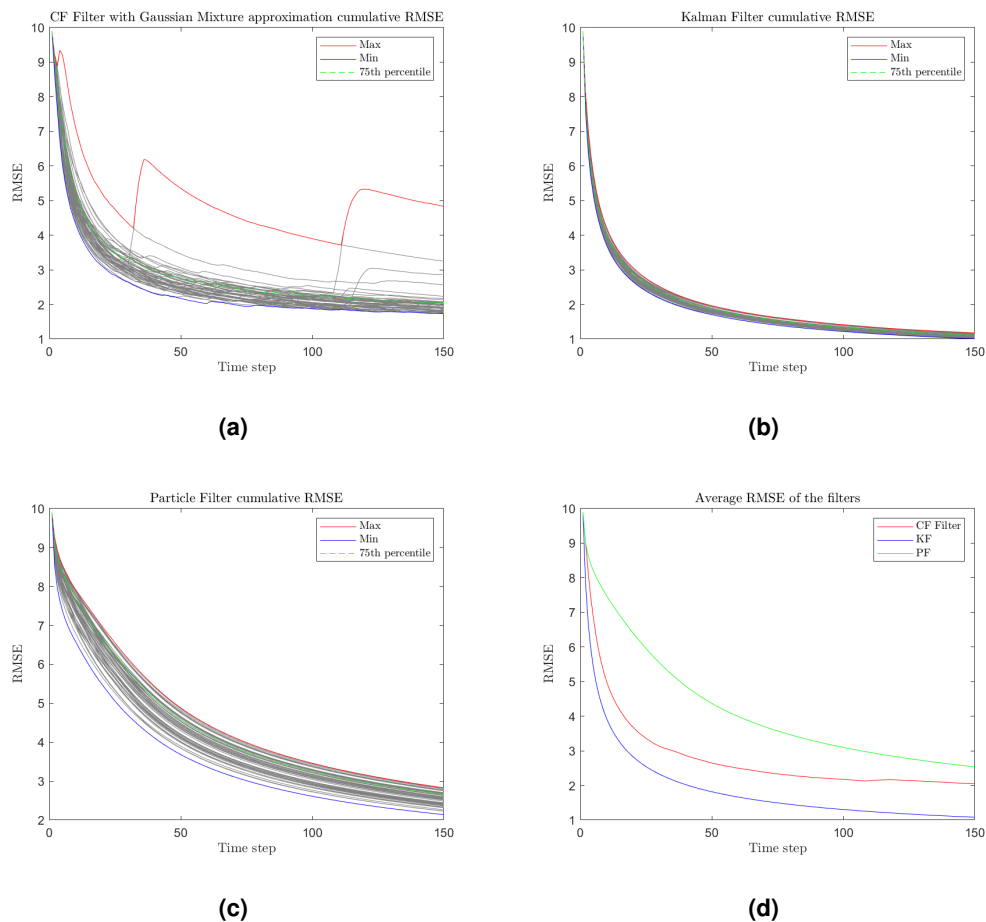
Despite these fluctuations, the CF Filter still demonstrates a strong average performance, as highlighted by the comparison in Figure 2.12(d). The average cumulative RMSE for the CF Filter remains competitive, particularly in comparison to the PF. This indicates that while individual runs may exhibit some instability due to the complexity of the Gaussian mixture approximation, the filter still approximates the state effectively over the 50 simulations.

In contrast, the KF, as shown in Figure 2.12(b), maintains a more stable performance. This is expected given that the Gaussian mixture noise used in these simulations can still be reasonably approximated by a zero-mean Gaussian distribution. The KF excels in these conditions, where its underlying assumptions about the noise model remain largely valid, resulting in smoother convergence and lower overall RMSE compared to the CF Filter. This performance reflects the KF's inherent design advantage, which enables it to handle Gaussian-like noise optimally, even when the noise model exhibits some additional complexity, as seen in the Gaussian mixture case.

The PF, however, continues to lag behind in terms of performance. As depicted in Figure 2.12(c), the PF shows slower convergence and a higher overall RMSE. The performance of the particle filter in this scenario may be hindered by the relatively low number of particles (256). The broader spread of error bounds observed in the PF results further reinforces its difficulty in effectively managing this scenario.

When comparing the average cumulative RMSE across all filters in Figure 2.12(d), the CF Filter, despite exhibiting fluctuations, still performs competitively with the KF. This suggests that while the

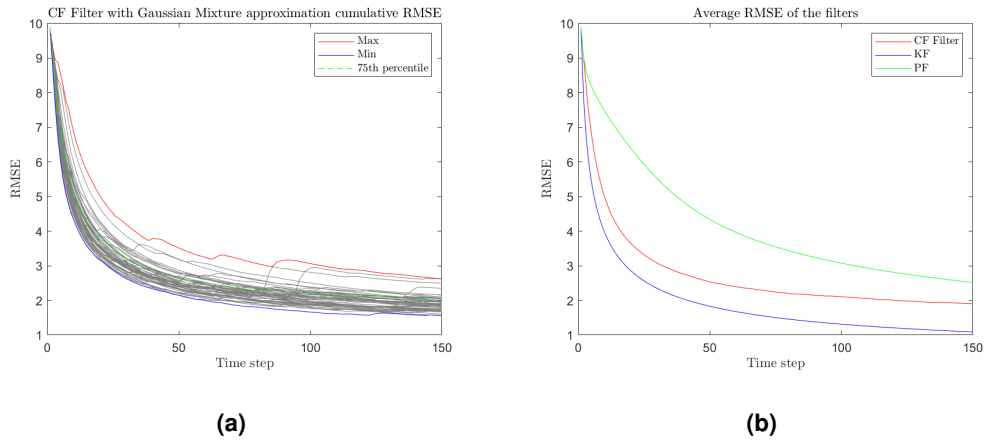
Gaussian mixture approximation introduces additional complexity, the CF Filter retains a solid overall performance.



**Figure 2.12:** Cumulative RMSE of the position of Gaussian Mixture w/ 2 peaks walk with Gaussian Mixture noise: a) CF Filter with 3 Gaussians; b) Kalman Filter; c) Particle Filter with 256 particles; d) Average of the cumulative RMSE over the 50 runs.

Increasing the number of Gaussians in the approximation from 3 to 4, as shown in Figures 2.13(a) and 2.13(b), results in a noticeable improvement in the stability of the CF Filter. The spread between the maximum and minimum error bounds tightens, and the overall convergence behavior becomes smoother compared to the 3-Gaussian case. This suggests that increasing the number of Gaussian components enhances the filter’s ability to capture the underlying complexity of the system dynamics and noise distributions, leading to more consistent state estimates.

In terms of the average cumulative RMSE, the CF Filter with 4 Gaussians shows a modest reduction in error relative to the 3-Gaussian scenario, as illustrated in Figure 2.13(b). This improvement helps to narrow the performance gap between the CF Filter and the KF, further emphasizing the value of a more refined Gaussian mixture approximation for accurately capturing the underlying noise characteristics.

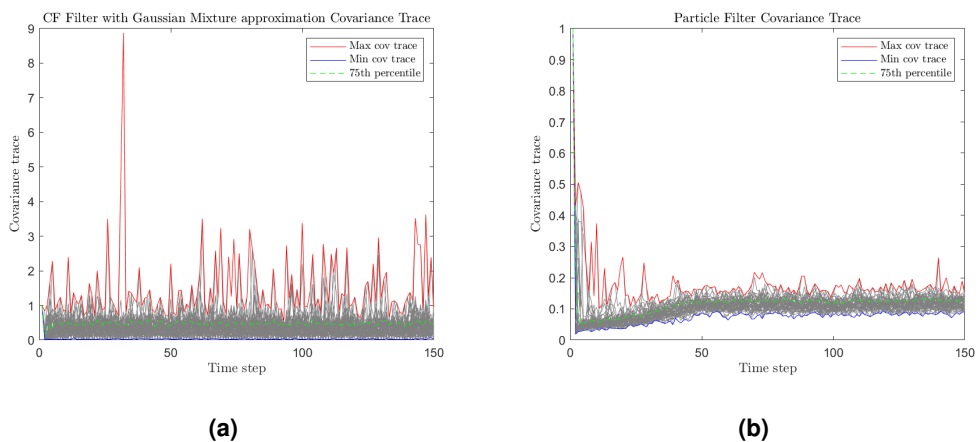


**Figure 2.13:** Cumulative RMSE of the position of Gaussian Mixture w/ 2 peaks walk with Gaussian Mixture noise: a) CF Filter with 4 Gaussians; b) Average of the cumulative RMSE over the 50 runs.

Figures 2.14(a) and 2.14(b) present the covariance matrix trace for the CF Filter with 3 and 4 Gaussians, respectively.

In the 3-Gaussian case, significant spikes in the covariance trace indicate periods of elevated uncertainty. Although the 75<sup>th</sup> percentile remains stable, the pronounced peaks in the maximum trace suggest that the filter occasionally struggles to accurately capture all modes of the noise distribution, resulting in increased uncertainty at specific time steps.

With 4 Gaussians, these spikes are less frequent and less pronounced. The overall maximum covariance trace is reduced, and the filter’s uncertainty is more consistently controlled. This reduction in uncertainty and the more consistent performance suggest that increasing the number of Gaussian components improves the filter’s ability to handle noise complexity effectively.



**Figure 2.14:** Covariance Matrix trace of the position estimate uncertainty in the Gaussian Mixture w/ 2 peaks walk with Gaussian Mixture noise: a) CF Filter with 3 Gaussians; b) CF Filter with 4 Gaussians.

To conclude the analysis of the Gaussian mixture approach, Table 2.2 provides a summary of the performance of each filter across the different simulation scenarios.

As shown in the table, the KF consistently outperforms the CF Filter across all noise scenarios, including those with exponential noise. Although the CF Filter demonstrates competitive performance, particularly when the number of Gaussian components is increased from 3 to 4, it still falls short of surpassing the KF. The observed variability in the CF Filter's performance can likely be attributed to the non-convex nature of the optimization problem inherent in the Gaussian mixture approximation, which may lead to convergence at local minima and suboptimal approximations of the system state. Furthermore, the number of Gaussian components employed may still be insufficient to fully capture the complexity of the state distribution, especially in cases involving intricate noise types such as exponential noise. This limitation is reflected in the higher RMSE values and greater variability compared to the KF.

In summary, while increasing the number of Gaussian components from 3 to 4 results in noticeable improvements in the CF Filter's performance, challenges remain in consistently matching the performance of the KF. The results suggest that incorporating more Gaussian components or employing a more sophisticated optimization approach may be necessary to achieve better state approximations and further reduce the RMSE, particularly in non-Gaussian noise environments. Nonetheless, the CF Filter offers notable flexibility and remains competitive, especially as the number of Gaussian components increases.

|                  |                                    | <b>Simulation Scenario</b>                    | <b>CF Filter w/ 3 Gaussian peaks</b> | <b>CF Filter w/ 4 Gaussian peaks</b> | <b>KF</b>     | <b>PF w/ 256 particles</b> |
|------------------|------------------------------------|---|--------------------------------------|--------------------------------------|---------------|----------------------------|
| <b>Mean RMSE</b> | <b>Average over 50 simulations</b> | G. Walk w/ G. Noise                           | 1.9380                               | 1.8374                               | <b>1.0595</b> | 2.5732                     |
|                  |                                    | G. Walk w/ GM Noise                           | 1.9860                               | 1.8288                               | <b>1.1021</b> | 2.5816                     |
|                  |                                    | G. Walk w/ Exp. Noise                         | 2.5700                               | 2.6360                               | <b>1.4468</b> | 3.3789                     |
|                  |                                    | GM w/ 2 peaks Walk w/ G. Noise                | 1.9336                               | 1.8633                               | <b>1.0522</b> | 2.5832                     |
|                  |                                    | GM w/ 2 peaks Walk w/ GM Noise                | 2.0531                               | 1.9073                               | <b>1.0907</b> | 2.5199                     |
|                  |                                    | GM w/ 2 peaks Walk w/ Exp. Noise              | 2.5323                               | 2.4467                               | <b>1.3974</b> | 3.4502                     |
|                  |                                    | GM w/ 3 peaks Walk w/ G. Noise                | 1.9284                               | 1.8785                               | <b>1.0572</b> | 2.5621                     |
|                  |                                    | GM w/ 3 peaks Walk w/ GM Noise                | 2.0566                               | 1.9670                               | <b>1.0913</b> | 2.5201                     |
|                  |                                    | GM w/ 3 peaks Walk w/ Exp. Noise              | 2.6267                               | 2.4170                               | <b>1.4407</b> | 3.8082                     |
|                  |                                    | <b>Standard deviation over 50 simulations</b> | G. Walk w/ G. Noise                  | 0.1905                               | 0.2026        | <b>0.0450</b>              |
|                  | G. Walk w/ GM Noise                | 0.2991  | 0.1551                               | <b>0.0427</b>                        | 0.1573        |                            |
|                  | G. Walk w/ Exp. Noise              | 0.4495  | 0.8652                               | <b>0.0488</b>                        | 0.2524        |                            |
|                  | GM w/ 2 peaks Walk w/ G. Noise     | 0.1473  | 0.1929                               | <b>0.0423</b>                        | 0.1609        |                            |
|                  | GM w/ 2 peaks Walk w/ GM Noise     | 0.4822  | 0.2389                               | <b>0.0386</b>                        | 0.1531        |                            |
|                  | GM w/ 2 peaks Walk w/ Exp. Noise   | 0.4619  | 0.4532                               | <b>0.0480</b>                        | 0.2143        |                            |
|                  | GM w/ 3 peaks Walk w/ G. Noise     | 0.1880  | 0.1878                               | <b>0.0457</b>                        | 0.1836        |                            |
|                  | GM w/ 3 peaks Walk w/ GM Noise     | 0.2693  | 0.4473                               | <b>0.0448</b>                        | 0.1809        |                            |
|                  | GM w/ 3 peaks Walk w/ Exp. Noise   | 0.4756  | 0.3312                               | <b>0.0512</b>                        | 0.3872        |                            |

**Table 2.2:** RMSE of all filters along different simulation scenarios, Gaussian Mixture approach.

### 2.5.2.C Approaches Comparison

When comparing the Dirac approach and the Gaussian mixture approach for hybrid filtering, it becomes clear that the Dirac method consistently yields superior and more stable results across the tested scenarios. The Dirac approach demonstrates robustness and adaptability, performing competitively with the KF in Gaussian noise settings and even surpassing it in non-Gaussian scenarios, such as those involving exponential noise. In contrast, the Gaussian mixture approach, while theoretically flexible, suffers from performance variability, particularly in more complex noise environments.

Beyond performance, the Dirac approach also offers significant computational efficiency. This is primarily due to the nature of the optimization problem solved during the prediction step. In the Dirac approach, the prediction involves approximating the CF using a finite sum of Dirac distributions, which reduces to a constrained linear least squares problem. Such problems are computationally simpler and faster to solve compared to the Gaussian mixture approach, which requires optimizing both the weights and parameters (means and covariances) of the Gaussian components. The non-convexity of the Gaussian mixture optimization problem further adds computational complexity, as it involves solving for multiple parameters and can become trapped in local minima. As the number of Gaussian components increases, the computational burden grows significantly, making the Dirac approach more efficient, particularly in resource-constrained environments.

In summary, the Dirac approach not only provides superior and more consistent results across various noise scenarios but also achieves this at a lower computational cost. This makes it a more practical choice for real-world applications where both accuracy and computational efficiency are critical. Conversely, while the Gaussian mixture approach offers theoretical flexibility, it may require more sophisticated optimization techniques or a larger number of Gaussian components to achieve comparable reliability and performance.

# 3

## Multi-Sensor State Estimation

### Contents

---

|   |    |
|---|----|
| 3.1 Related Work . . . . .                        | 43 |
| 3.2 Multi-Sensor Problem . . . . .                | 46 |
| 3.3 Background Technical Details . . . . .        | 47 |
| 3.4 Proposed Approach . . . . .                   | 51 |
| 3.5 Experimental Results and Discussion . . . . . | 53 |

---

Chapter 3 addresses the problem of SE in scenarios involving multiple sensors. Unlike single-sensor SE, the multi-sensor approach capitalizes on a network of sensors working collaboratively to estimate the state of a dynamic system, thereby improving robustness, accuracy, and resilience. This chapter delves into the complexities of fusing information from multiple sensors, particularly under both Gaussian and non-Gaussian noise assumptions. Methods such as Distributed Sensor Fusion based on Average Consensus and the WB framework for sensor fusion are examined through detailed theoretical analysis and simulations.

## **3.1 Related Work**

This section provides a comprehensive review of the existing literature and methodologies in the domain of multi-sensor SE, along with an exploration of practical applications of WB in SE techniques.

### **3.1.1 Multi-Sensor Gaussian Assumption**

In the field of SE for linear systems, addressing the MA-GA presents unique challenges. This subsection focuses on scenarios where multiple sensors operate under Gaussian noise assumptions. Traditional single-sensor methods like the KF, while effective under Gaussian conditions, face significant complexities when extended to multi-sensor systems. This section examines advanced methodologies designed to efficiently integrate data from multiple sensors and addresses the unique challenges associated with MA-GA in SE.

#### **3.1.1.A Distributed Sensor Fusion Based on Average Consensus**

To understand the implementation of distributed sensor fusion based on average consensus, it is essential to first grasp the concept of average consensus, as it underpins the algorithm's operation. The principle of average consensus is a process by which a network of agents, or sensor nodes, reaches agreement on the average value of their individual measurements or states without centralized control. Essentially, each node iteratively updates its current value by incorporating the values of its immediate neighbors. The convergence of this iterative process ensures that all nodes in the network ultimately agree on the average value of the collective initial states. This method is particularly advantageous in distributed systems, as it enables nodes to arrive at a collective decision using only local information, thereby enhancing both robustness and scalability.

The distributed sensor fusion algorithm based on average consensus presented in [12] addresses the challenges inherent in networks of distributed sensors, where each sensor takes linear measurements of unknown parameters under independent Gaussian noise conditions. The focus is on a distributed

iterative scheme that does not require direct point-to-point message passing. Instead, it leverages the principle of average consensus across the network, enabling the computation of the maximum-likelihood estimate of parameters in a robust and efficient manner.

Each sensor node maintains a fixed and small storage memory, utilizing the same data structure across all nodes, and exchanges information with its neighbors in a uniform and simple manner. This exchange involves updating each node's data with a weighted average of its neighbors' data. Consequently, each node can compute a local weighted least-squares estimate of parameters, which, over time, converges to the global maximum-likelihood solution.

This scheme is particularly well-suited for networks with dynamically changing topologies and unreliable communication links. It functions effectively even when communication links are intermittent, as long as the network remains connected over the long term. This robustness to communication failures and network topology changes makes it ideal for real-world applications, where such challenges are commonplace.

From a practical standpoint, each sensor node maintains a local composite information matrix and a local composite information state. Throughout the iterative process, nodes achieve average consensus by adjusting their local composite information matrix and state based on data received from neighboring nodes. This process does not require complex message routing but relies instead on the simple diffusion of information through average consensus. As a result, each node independently and accurately computes the maximum-likelihood estimate, thereby contributing to the overall robustness and efficiency of the sensor network.

### **3.1.1.B Distributed Kalman Filtering**

In [11] and [13], the authors explore the problem of DKF in sensor networks, addressing various aspects and implementations of this method. Both papers utilize the information filter form of the KF to formulate the DKF scheme. The first paper, [11], introduces an approach that incorporates embedded filters of consensus-based fusion of sensory data, proposing low-pass and band-pass consensus filters for implementing the DKF.

The second paper, [13], extends this exploration by proposing a high-pass consensus filter. However, the key contribution lies in presenting an algorithm where the consensus mechanism is based on estimates derived from local Kalman filtering (LKF). This subsection primarily focuses on this latter algorithm.

The author in [13] presents the Kalman-Consensus Filter, which integrates the principles of the KF with a consensus mechanism. This integration is essential for ensuring consistent state estimates across a distributed sensor network.

The algorithm builds upon the concept of LKF, where each sensor node uses only its own measure-

ments and those from its immediate neighbors for state estimation. While LKF provides a foundational level of distributed state estimation, it lacks an explicit mechanism for achieving consensus across all nodes in the network, potentially leading to divergence in state estimates and inaccurate or unreliable system-level state estimation.

The Kalman-Consensus Filter addresses these shortcomings by incorporating a consensus term into the state estimation process of each sensor node. This term plays a crucial role in aligning the individual estimates of each sensor with those of its neighbors, thus ensuring a more unified and accurate state estimation across the entire network.

One of the defining features of the Kalman-Consensus Filter is its capacity to ensure cohesion and stability in the state estimation process. By effectively reducing the discrepancies among sensor estimates, the filter guarantees that all sensors within the network asymptotically converge to a consistent and accurate estimate of the target state.

### **3.1.2 Multi-Sensor Non-Gaussian Assumption**

In the context of SE for linear systems, the MA-NGA introduces distinct complexities. This subsection focuses on scenarios where sensor networks are subject to non-Gaussian noise, thereby challenging traditional sensor data fusion and estimation paradigms. Here, we explore methodologies specifically tailored to address the challenges inherent in MA-NGA within the SE framework. These techniques aim to effectively combine data from a network of sensors, each potentially exhibiting unique non-Gaussian noise characteristics, to achieve accurate and reliable state estimation.

#### **3.1.2.A Distributed State Estimation Based on Kullback–Leibler Average**

The authors in [16] introduce a novel approach to state estimation in sensor networks. In this framework, each node within the network, equipped with processing, communication, and sensing capabilities, engages in fusing local information with data acquired from neighboring nodes. This process is embedded within a Bayesian framework, where the fusion of information is guided by an information-theoretic perspective, specifically focusing on the KL average of local PDFs.

The core concept of this methodology is to define the average PDF as the one that minimizes the sum of information gains, quantified by KL divergences from the original sensor PDFs. This results in the KL average PDF being essentially the geometric mean of the sensor PDFs. The aim is to develop a consensus algorithm that allows all nodes within the network to reach agreement regarding the PDF of the estimated state. This consensus algorithm consists of two main components: the exchange of information between a node (sensor) and its neighbors, and the mechanism for updating the local PDF based on received information. The problem is formulated to determine a consensus algorithm such

that the PDF at each node asymptotically converges to a uniform KL average of the initial PDFs of all nodes.

## 3.2 Multi-Sensor Problem

The objective of this section is to formulate the state estimation problem from a multi-sensor perspective, providing a rigorous mathematical framework for the analysis conducted in this work.

SE in a multi-sensor scenario involves leveraging a network of distributed sensors, each independently collecting observations, to collaboratively estimate the state of a dynamic system. The overarching goal is to integrate the collective data from all sensors to achieve a more robust and accurate state estimate than that which could be attained by any individual sensor alone.

The problem is represented by an interconnected network of sensors, each acquiring observations over time. The network is modeled as a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V} = \{v_1, v_2, \dots, v_m\}$  denotes the set of  $m$  sensor nodes, and  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  represents the communication links among them.

- **State Equation:** The state equation remains identical to equation 2.1 from the single sensor scenario.
- **Observation Equation for Sensor  $i$ :** The observation model for sensor  $i$  at time  $t$  is given by:

$$z_t^{(i)} = H^{(i)} x_t + v_t^{(i)} \quad (3.1)$$

where  $z_t^{(i)} \in \mathbb{R}^{p_i}$  is the observation vector for sensor  $i$ ,  $H^{(i)} \in \mathbb{R}^{p_i \times n}$  is the observation matrix, and  $v_t^{(i)} \sim \mathcal{D}(\theta^{(i)})$  represents the measurement noise for sensor  $i$ .

This multi-sensor problem builds upon assumptions analogous to those in the single sensor case, such as system linearity, time invariance of system matrices, known parameters, and independent noise vectors. However, a distinctive feature of the multi-sensor context is the consideration of network topology. The sensor network is represented as a connected graph  $\mathcal{G}$ , ensuring that communication pathways exist among all sensor nodes. The analysis includes two observability conditions: individual sensor observability (i.e., each sensor's matrix pair  $(A, H^{(i)})$  is observable) and collective observability of the entire network. The latter condition is generally less restrictive and preferred for ensuring overall system observability. The ultimate aim is to integrate the observations  $z_t^{(i)}$  from all sensors to generate a more precise global state estimate  $\hat{x}_t$  than any individual sensor could achieve.

### 3.3 Background Technical Details

Section 3.3 delves into the technical details and mathematical formulations underlying the methodologies introduced in Section 3.1, with a focus on distributed sensor fusion and consensus-based estimation.

#### 3.3.1 Multi-Sensor Gaussian Assumption

##### 3.3.1.A Distributed Sensor Fusion Based on Average Consensus

In distributed sensor fusion based on average consensus, the goal is to estimate the maximum likelihood state,  $\hat{x}_{t_{ML}} \in \mathbb{R}^n$ , given the measurements  $z_t^{(i)} \in \mathbb{R}^{p_i}$  from  $m$  sensors observing the state  $x_t \in \mathbb{R}^n$ :

$$z_t^{(i)} = H^{(i)}x_t + v_t^{(i)}, \quad i = 1, \dots, m \quad (3.2)$$

Here,  $H^{(i)} \in \mathbb{R}^{p_i \times n}$  denotes the measurement matrix for sensor  $i$ , and  $v_t^{(i)} \in \mathbb{R}^{p_i}$  is the measurement noise, assumed to be Gaussian with zero mean and covariance matrix  $R^{(i)} \in \mathbb{R}^{p_i \times p_i}$ .

The algorithm initializes the local information matrix  $U_t^{(i)} \in \mathbb{R}^{n \times n}$  and the local information state  $q_t^{(i)} \in \mathbb{R}^n$  for sensor  $i$  at time  $t = 0$  as follows:

$$U_0^{(i)} = H^{(i)\top} R_t^{(i)-1} H^{(i)} \quad (3.3)$$

$$q_0^{(i)} = H^{(i)\top} R_t^{(i)-1} z_0^{(i)} \quad (3.4)$$

The initial state captures each sensor's own measurement information and its uncertainty via the inverse of the measurement noise covariance.

The fusion process is carried out iteratively, updating  $U_t^{(i)}$  and  $q_t^{(i)}$  using the average consensus protocol. At each iteration  $t$ , the update rules are as follows:

$$U_{t+1}^{(i)} = W_t^{ii} U_t^{(i)} + \sum_{j \in N_t^{(i)}} W_t^{ij} U_t^{(j)} \quad (3.5)$$

$$q_{t+1}^{(i)} = W_t^{ii} q_t^{(i)} + \sum_{j \in N_t^{(i)}} W_t^{ij} q_t^{(j)} \quad (3.6)$$

Here,  $N_t^{(i)}$  represents the set of neighboring nodes for sensor  $i$  at time  $t$ , and  $W_t^{ij}$  are the weights assigned either using the maximum-degree or Metropolis weighting strategy, here the superscripts  $ij$  in  $W$  refer to the entries in the weight matrix  $W$ . The weights  $W_t^{ij}$  are defined as follows:

$$W_t^{ij} = \begin{cases} \frac{1}{1 + \max\{d_t^{(i)}, d_t^{(j)}\}}, & \text{if } \{i, j\} \in \mathcal{E}_t \\ 1 - \sum_{k \in N_t^{(i)}} W_t^{ik}, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \quad (3.7)$$

In these definitions,  $d_t^{(i)}$  represents the degree of node  $i$  at time  $t$ , and  $\mathcal{E}_t$  represents the set of edges at time  $t$ . The maximum-degree weights require knowledge of the total number of nodes, whereas Metropolis weights require only local degree information.

Through iterative updates, each node's local estimate converges to the global maximum-likelihood estimate,  $\hat{x}_{t,ML}$ , without the need for centralized coordination. The distributed nature of this approach ensures robustness to link failures and dynamic changes in network topology, provided the network remains connected. Intermediate state estimates are available at node  $i$  as soon as  $U_t^{(i)}$  becomes invertible:

$$\hat{x}_t^{(i)} = U_t^{(i)-1} q_t^{(i)}, \quad i = 1, \dots, m \quad (3.8)$$

### 3.3.1.B Distributed Kalman Filtering

The Kalman-Consensus Filter is an extension designed for distributed state estimation in sensor networks, where each sensor node collaboratively contributes to the estimation process. The state-space model employed in this approach is as follows:

Consider a state  $x_t \in \mathbb{R}^n$  that evolves over time according to the linear state-space model:

$$x_t = Ax_{t-1} + d_t, \quad (3.9)$$

where  $A \in \mathbb{R}^{n \times n}$  represents the state transition matrix, and  $d_t \in \mathbb{R}^n$  denotes the process noise, which is assumed to be Gaussian with zero mean and covariance matrix  $Q \in \mathbb{R}^{n \times n}$ .

Each sensor  $i$  in the network makes a measurement  $z_t^{(i)} \in \mathbb{R}^{p_i}$  of the state  $x_t$ , given by:

$$z_t^{(i)} = H^{(i)} x_t + v_t^{(i)}, \quad i = 1, \dots, m, \quad (3.10)$$

where  $H^{(i)} \in \mathbb{R}^{p_i \times n}$  is the measurement matrix, and  $v_t^{(i)} \in \mathbb{R}^{p_i}$  represents the measurement noise, assumed to be Gaussian with zero mean and covariance matrix  $R_t^{(i)} \in \mathbb{R}^{p_i \times p_i}$ .

The Kalman-Consensus Filter updates the state estimate and error covariance for each sensor node  $i$  by computing the consensus information state  $q_t^{(j)}$  and information matrix  $U_t^{(j)}$  for each neighboring sensor  $j$  as follows:

$$q_t^{(j)} = H^{(j)\top} R_t^{(j)-1} z_t^{(j)}, \quad \forall j \in J_i, \quad (3.11)$$

$$U_t^{(j)} = H^{(j)\top} R_t^{(j)-1} H^{(j)}, \quad \forall j \in J_i, \quad (3.12)$$

where  $J_i$  is the set consisting of sensor  $i$  and its neighbors. Sensor  $i$  then aggregates the information state  $y_t^{(i)}$  and the information matrix  $S_t^{(i)}$  from its own measurements and those of its neighbors:

$$y_t^{(i)} = \sum_{j \in J_i} q_t^{(j)}, \quad (3.13)$$

$$S_t^{(i)} = \sum_{j \in J_i} U_t^{(j)}. \quad (3.14)$$

Sensor  $i$  then updates its state estimate  $\hat{x}_t^{(i)}$  using the Kalman gain  $M_t^{(i)}$  and a consensus term:

$$M_t^{(i)} = (P_t^{(i)-1} + S_t^{(i)})^{-1}, \quad (3.15)$$

$$\hat{x}_t^{(i)} = \bar{x}_t^{(i)} + M_t^{(i)} (y_t^{(i)} - S_t^{(i)} \bar{x}_t^{(i)}) + \epsilon M_t^{(i)} \sum_{j \in N_t^{(i)}} (\bar{x}_t^{(j)} - \bar{x}_t^{(i)}), \quad (3.16)$$

where  $N^{(i)}$  denotes the set of neighbors of sensor  $i$ ,  $\bar{x}_t^{(i)}$  is the prior state estimate of node  $i$ , and  $\epsilon$  is a small positive constant representing the consensus strength.

The state estimate and error covariance matrix are then updated for the next time step:

$$P_{t+1}^{(i)} = A M_t^{(i)} A^\top + Q, \quad (3.17)$$

$$\bar{x}_{t+1}^{(i)} = A \hat{x}_t^{(i)}. \quad (3.18)$$

### 3.3.2 Multi-Sensor Non-Gaussian Assumption

#### 3.3.2.A Distributed State Estimation Based on Kullback–Leibler Average

The distributed state estimation approach based on the KL Average in sensor networks utilizes a consensus mechanism where each node in the network, equipped with a local PDF of the state estimate, collaborates to achieve a unified estimate of the state. At the core of this process is the weighted KL Average, which is defined as follows:

The weighted KL Average is represented as the normalized geometric mean of the sensor PDFs

$p^{(i)}(x)$ . It is formulated by the following equation:

$$p(x) = \frac{\prod_{i \in N} [p^{(i)}(x)]^{w^{(i)}}}{\int \prod_{i \in N} [p^{(i)}(x)]^{w^{(i)}} dx} \quad (3.19)$$

where  $p^{(i)}(x)$  denotes the local PDFs at each sensor node,  $w^{(i)}$  are the corresponding weights, and  $N$  represents the set of nodes in the network.

For the distributed consensus algorithm, two fundamental operators that act on PDFs are defined:

More specifically, given two PDFs  $p(\cdot)$  and  $q(\cdot)$ , their addition is defined as:

$$p(x) \oplus q(x) = \frac{p(x)q(x)}{\int p(x)q(x)dx} \quad (3.20)$$

For a given positive scalar  $w$  and a PDF  $p(x)$ , the scalar multiplication operator is defined as:

$$w \odot p(x) = \frac{[p(x)]^w}{\int [p(x)]^w dx} \quad (3.21)$$

where  $[\cdot]^w$  represents exponentiation of  $\cdot$  to the power  $w$ .

The consensus algorithm updates the PDF at each node  $i$  in the network, denoted as  $p_{\ell+1}^{(i)}(x)$ , by integrating information from its neighbors. The update rule is given by:

$$p_{\ell+1}^{(i)}(x) = \bigoplus_{j \in N^{(i)}} \left( w^{ij} \odot p_{\ell}^{(j)}(x) \right), \quad \forall i \in N \quad (3.22)$$

where  $N^{(i)}$  represents the set of neighbors of node  $i$ ,  $w^{ij}$  are the non-negative weights that sum to 1 for each node  $i$ , with the superscripts  $ij$  in  $w$  referring to entries in the weight matrix  $w$ , and  $\ell$  represents the iteration of the consensus algorithm.

As the number of consensus steps increases, each local PDF gradually converges to the collective KL Average. Convergence is guaranteed when the consensus weights  $w^{ij}$  ensure that the consensus matrix is both primitive and doubly stochastic, thereby leading to a uniform distribution of weights across the network in the limit.

### 3.3.3 Wasserstein Barycenter

The WB is a concept from optimal transport theory that provides a systematic approach to aggregate multiple probability distributions into a single representative distribution. Analogous to the classical mean in Euclidean space, the WB can be interpreted as the 'mean' of probability distributions in the space of probability measures, while inherently accounting for the underlying geometry of those distributions.

In a more general sense, the WB aims to identify a central distribution that minimizes divergence from a set of given distributions in the Wasserstein metric sense. This objective resembles that of computing

an average, but instead of averaging scalar values, the WB averages distributions, considering the 'distance' between them.

Consider a set of probability distributions  $\{\mu_i\}_{i=1}^N$ . The objective is to find a barycenter distribution  $\nu$  that minimizes the sum of Wasserstein distances to each  $\mu_i$ , which is mathematically defined as follows:

$$\nu^* = \arg \min_{\nu \in \mathcal{P}(\mathbb{R}^n)} \sum_{i=1}^N \lambda_i \Phi_2^2(\mu_i, \nu), \quad (3.23)$$

where  $\mathcal{P}(\mathbb{R}^n)$  denotes the space of probability measures on  $\mathbb{R}^n$ ,  $\Phi_2(\mu_i, \nu)$  represents the Wasserstein-2 distance between distributions  $\mu_i$  and  $\nu$ , and  $\lambda_i$  are the weights associated with each distribution, satisfying  $\sum_{i=1}^N \lambda_i = 1$  and  $\lambda_i \geq 0$ . The Wasserstein-2 distance is given by:

$$\Phi_2^2(\mu_i, \nu) = \inf_{\gamma \in \Gamma(\mu_i, \nu)} \int_{\mathbb{R}^n \times \mathbb{R}^n} \|x - y\|_2^2 d\gamma(x, y), \quad (3.24)$$

where  $\Gamma(\mu_i, \nu)$  is the set of all probability measures on  $\mathbb{R}^n \times \mathbb{R}^n$  that have marginals  $\mu_i$  and  $\nu$ , and  $\|x - y\|_2$  denotes the Euclidean norm in  $\mathbb{R}^n$ .

The WB framework provides a powerful method for data fusion and state estimation in multi-sensor systems, particularly under non-Gaussian noise conditions, as it enables aggregation while respecting the intrinsic geometries of the distributions involved.

## 3.4 Proposed Approach

This section outlines our proposed solution to the multi-sensor state estimation problem defined in Section 3.2.

### 3.4.1 Wasserstein Barycenter for Sensor Fusion

The motivation for employing the WB in sensor fusion arises from the intrinsic complexity of integrating data from multiple sensors, each characterized by distinct probability distributions. Conventional fusion methods often assume homogeneity among these distributions, which can lead to suboptimal performance, especially when sensor data exhibit diverse, non-Gaussian characteristics.

The WB provides a robust solution by offering a mechanism for merging heterogeneous distributions into a unified representative distribution. This approach is particularly beneficial in scenarios involving sensor data with varied noise profiles, as it respects the underlying geometry of the distributions rather than relying on simplistic averaging techniques.

The central premise of this approach is to combine information from each sensor in a way that minimizes the overall divergence between the fused distribution and the individual sensor distributions.

Unlike traditional methods that aggregate scalar values, the WB framework aggregates the entire distributions, ensuring that the final estimate captures the geometric relationships between the sensor data.

By leveraging the WB for sensor fusion, we can effectively aggregate data from multiple sources while preserving the fundamental characteristics of each individual distribution. This ensures that the fused result is not simply an aggregate, but a well-balanced representation that minimizes discrepancies among the original distributions. The optimization inherent in the WB computation also provides flexibility in weighting each sensor's contribution, thereby ensuring that the resulting distribution reflects the relative importance of each sensor's data.

Consider a scenario involving  $N$  state estimates, each generated by a distinct sensor. These sensors may be positioned in different locations or exposed to varying noise conditions, leading to different probability distributions for their respective state estimates. In such circumstances, directly combining these estimates using conventional methods could yield inaccurate results, as these methods do not account for the individual characteristics of each distribution.

Instead, each sensor transmits its local probability distribution or state estimate to a central computational unit. This unit, using the WB framework, fuses the received distributions by determining a central distribution that best represents the combined information from all sensors. The resulting PDF, derived through the WB, minimizes the Wasserstein distance between the central estimate and the individual sensor distributions, thereby ensuring that the fused result appropriately balances the contributions from each sensor.

Once the fused PDF is derived, the central unit communicates this updated distribution back to each sensor, allowing each sensor to refine its own state estimate based on the fused information. This process ensures that each sensor benefits from the collective data of the entire network, ultimately enhancing individual estimates.

Beyond the centralized approach, the WB can also be computed in a distributed manner [18], thereby improving both scalability and robustness, particularly in large sensor networks. A fully centralized approach, while effective, can become a bottleneck in large-scale systems due to its reliance on a single processing unit for computation and information dissemination.

In contrast, a distributed computation of the WB allows each sensor, or groups of sensors, to locally compute partial updates. These updates are then exchanged with neighboring nodes or segments of the network, thus reducing communication overhead and distributing the computational burden across the network. This decentralized approach not only scales more efficiently as the number of sensors grows but also enhances the network's resilience by removing single points of failure.

In a distributed framework, sensors collaborate by exchanging intermediate estimates, progressively refining the fused PDF without requiring centralized coordination. Over several iterations, the network converges to a global WB that encapsulates the collective information from all sensors. This makes the

WB framework particularly suitable for real-time applications in dynamic environments where scalability, robustness, and efficiency are paramount.

To further illustrate this approach, consider the emergence of the WB when the state estimates from two distinct sensors correspond to two different Gaussian distributions. As the algorithm for calculating the WB iterates, with iterations increasing from Sub-figure 3.1(a) to Sub-figure 3.1(c) in Figure 3.1, we observe the gradual formation of a barycenter distribution that effectively represents the WB. This example highlights the WB's capability to merge disparate distributions while preserving essential characteristics, thereby yielding a meaningful and balanced fused distribution.

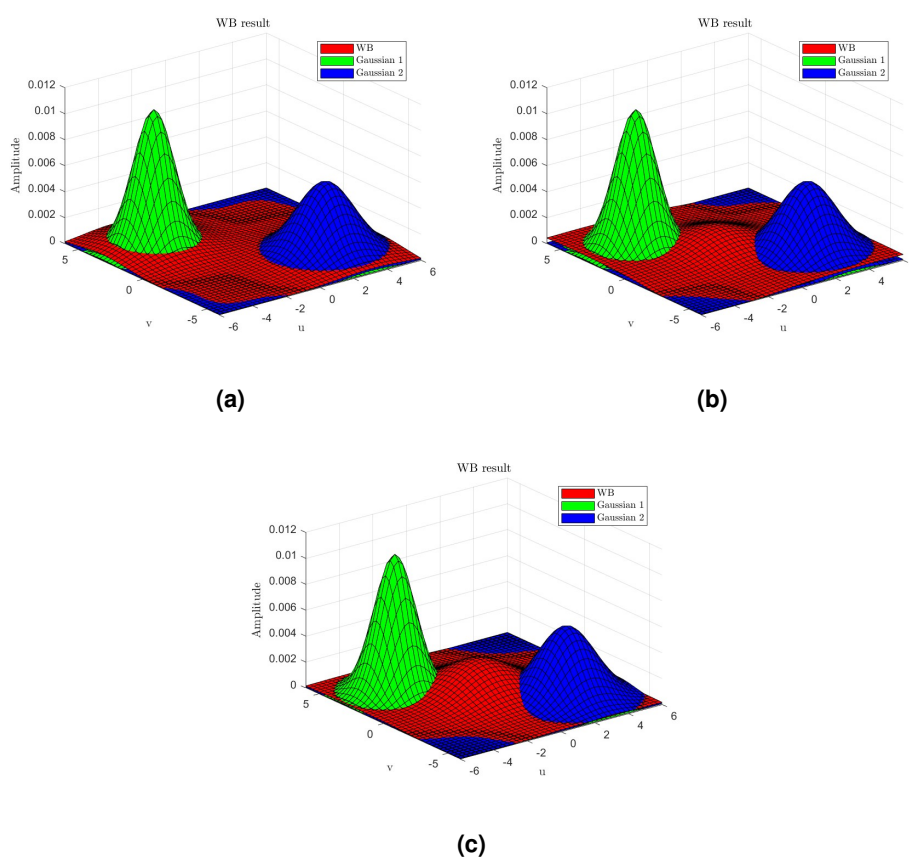


Figure 3.1: Wasserstein Barycenter along increasing iterations.

### 3.5 Experimental Results and Discussion

This section presents the results of the simulations conducted to address the multi-sensor problem, with a particular emphasis on the application of the WB framework. Details of the simulations are provided in Subsection 3.5.1, followed by an analysis of the corresponding results in Subsection 3.5.2.

### 3.5.1 Simulation

The simulations were designed to evaluate two distinct scenarios: one where the sensor measurement noise is Gaussian, and another where different sensors are affected by heterogeneous noise types. To simplify the analysis, both simulations were performed on a network comprising three sensors with mutual communication capabilities. Each sensor was capable of executing local filtering operations, and the resulting local estimates were subsequently communicated to a centralized unit, which fused all the information using the WB framework (see Figures 3.2 and 3.3). Thus, the implemented algorithm functions in a centralized manner.

In all simulations, the system followed discrete double integrator dynamics with a sampling time of  $T_s = 0.1$  s. The model equations are analogous to equations 2.1 and 3.1, with the key difference being that the disturbance term  $d_t$  is omitted in these simulations. Regarding the observation model, the observation matrices  $H^{(i)}$  are assumed to be identical for all sensors:

$$H^{(i)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \forall i \in \mathcal{V} \quad (3.25)$$

In both scenarios, the objective is to estimate the position of a vehicle following a two-dimensional trajectory in the shape of a figure-eight. The initial position of the vehicle is not precisely known and is assumed to be  $[0 \ 0]^\top$  with some associated uncertainty, allowing for the estimation of an initial PDF.

The simulations operate under several additional assumptions. In the second scenario, where each sensor is affected by distinct types of noise, the local filters are configured based on the individual assumptions each sensor makes about the noise affecting its measurements. Specifically, the first sensor assumes GMM noise, the second sensor assumes Gaussian noise, and the third sensor is affected by uniform noise. Accordingly, the first sensor employs a Gaussian mixture filter, the second sensor utilizes a KF, and the third sensor implements an interval observer. In the first scenario, where the noise affecting all sensors is Gaussian, all sensors employ local KFs.

Once the local filtering operations are completed, the WB is applied to fuse the information and produce a final state estimate. To provide a thorough comparison, three additional algorithms were implemented: a centralized KF, the average consensus algorithm described in Section 3.1.1.A, and a straightforward algorithm that averages the position estimates from each individual sensor. The first scenario, involving Gaussian noise, serves as a baseline since the centralized KF operates under optimal conditions in this context.

Both scenarios were simulated 30 times, with each simulation comprising 250 iterations (time steps). In each of these 30 runs, the initial noise distributions were generated randomly. To evaluate performance, the following metrics were used: the RMSE, calculated over each of the 30 runs; the  $\ell^2$ -norm of the position error vector at each time step, defined as  $e_t = x_t - \hat{x}_t$ , for a representative simulation; and

the trace of the covariance matrix, used to quantify the uncertainty of the estimate for a representative simulation.

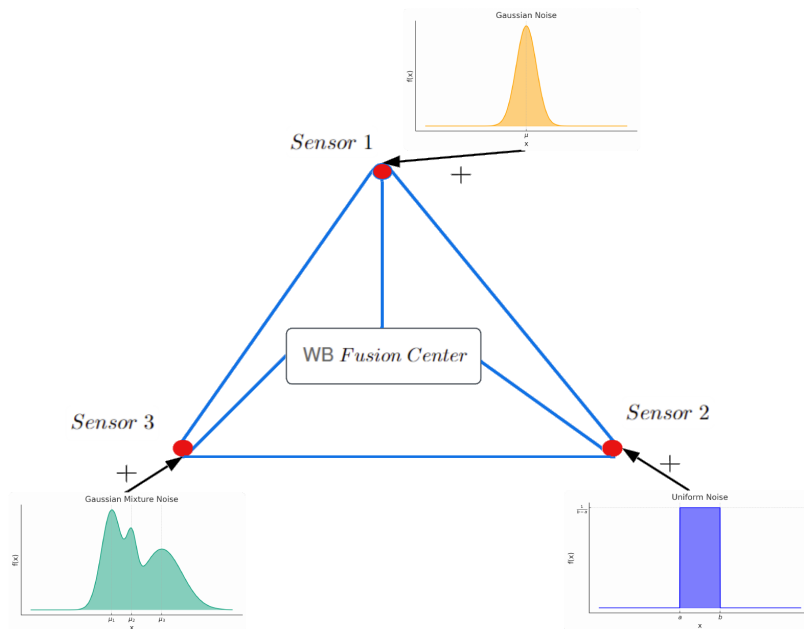


Figure 3.2: Sensor network example.

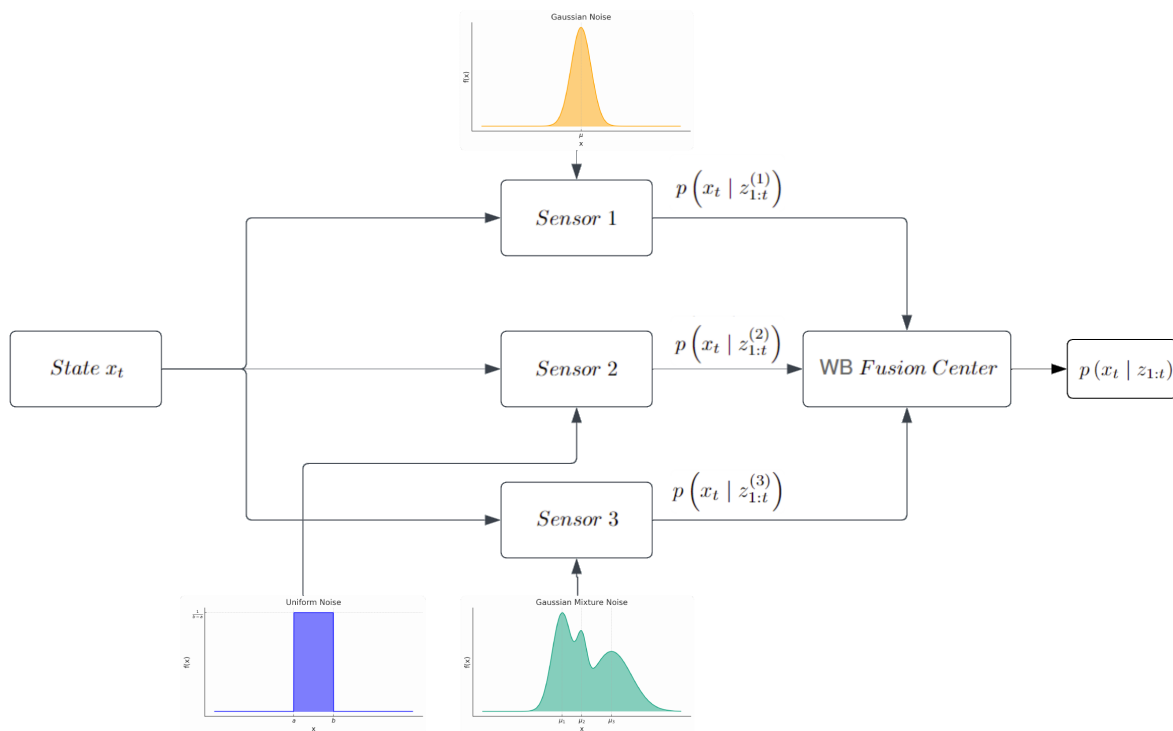


Figure 3.3: Information fusion diagram.

## 3.5.2 Results

In this subsection, we present a detailed analysis of the simulation results discussed previously. The results are illustrated in Figures 3.4 and 3.5, where Figures 3.4(a), 3.4(b), 3.5(a), and 3.5(b) display their y-axes on a logarithmic scale to highlight variability across orders of magnitude.

### 3.5.2.A Gaussian Noises Simulation

Figures 3.4(a) and 3.4(b) demonstrate that while the WB fusion method achieves highly accurate state estimates, it does so with noticeably higher uncertainty compared to the centralized KF.

In Figure 3.4(b), the covariance trace for the WB fusion does not exhibit the consistent downward trend seen with the centralized KF. This discrepancy can be explained by the fundamental differences between the two fusion methodologies.

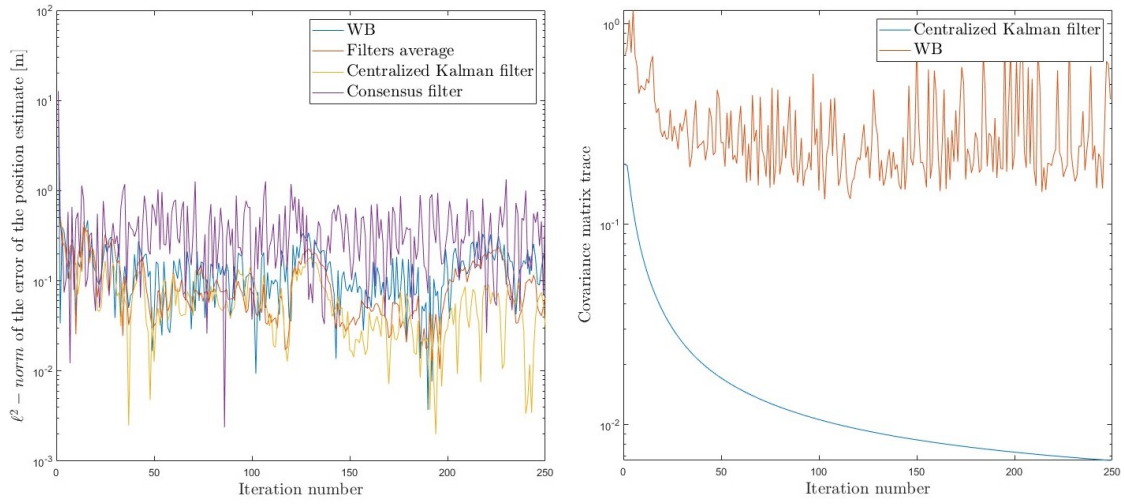
The WB algorithm aggregates probability distributions from multiple sensors, yielding an optimal central estimate in the Wasserstein sense. However, unlike the KF, the WB does not explicitly aim to iteratively minimize uncertainty. The KF achieves a systematic reduction in uncertainty over time through repeated measurement incorporation and update using an optimal gain (Kalman Gain), which ensures a gradual reduction in the covariance trace as confidence in the estimate grows.

In contrast, the WB fusion process aggregates sensor distributions based on their geometric relationships in Wasserstein space rather than following a recursive uncertainty reduction strategy. Consequently, fluctuations in uncertainty arise due to discrepancies between individual sensor estimates, particularly when sensors are subject to differing noise realizations or varying degrees of confidence.

Moreover, the WB fusion process tends to be more sensitive to inconsistencies among sensor distributions. When individual sensor estimates deviate significantly due to random noise, the resulting fused distribution can exhibit increased uncertainty. This sensitivity explains the irregular behavior of the covariance trace over time, as the WB continuously merges these disparate estimates rather than converging to a single optimal solution like the KF.

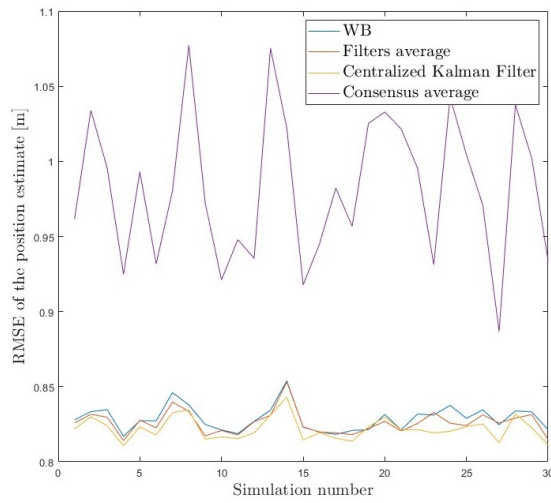
As depicted in Figure 3.4(c) and Table 3.1, the centralized KF achieves the lowest overall RMSE, operating optimally under Gaussian noise conditions. However, the WB fusion demonstrates comparable accuracy, closely matching both the centralized KF and the simple averaging of local filter outputs.

It is noteworthy that although the WB fusion does not outperform the centralized KF in accuracy, its performance remains within an acceptable range, highlighting its capability for reliable state estimation in a centralized fusion context. The greater variability in RMSE observed for the WB relative to the KF underscores its nature as a general-purpose fusion method—one not specifically optimized for Gaussian noise, yet still effective in such contexts.



(a)

(b)



(c)

**Figure 3.4:** Gaussian noises simulation: a)  $\ell^2$ -norm of the position error; b) Covariance matrix trace of the position uncertainty; c) RMSE over multiple simulations.

|                             | WB     | Filters average | Centralized KF | Consensus average |
|-----------------------------|--------|-----------------|----------------|-------------------|
| Mean RMSE [m]               | 0.8288 | 0.8261          | 0.8220         | 0.9819            |
| RMSE Standard Deviation [m] | 0.0084 | 0.0079          | 0.0076         | 0.0491            |

**Table 3.1:** RMSE of Gaussian noise simulation.

### 3.5.2.B Different Noises Simulation

In the second scenario, where sensors are subject to different types of noise, the WB fusion method demonstrates significant adaptability to more complex conditions. The heterogeneous noise across the sensor network presents challenges that traditional methods, such as the centralized KF, are less equipped to handle, as evidenced by the superior performance metrics of the WB fusion.

Figure 3.5(a) shows that the WB fusion exhibits a clear advantage in terms of position error, particularly during early iterations. Although the centralized KF maintains reasonable accuracy, it struggles with the variability in noise across sensors, resulting in increased errors. In contrast, the WB effectively aggregates sensor data in the Wasserstein sense, leading to a consistently lower error profile.

The simple averaging approach for filter outputs also achieves competitive position error results; however, it is fundamentally limited in that it provides only a single state estimate vector, representing the mean of individual sensor estimates. This approach lacks crucial information about the uncertainty or probabilistic characteristics of the estimate. Thus, despite achieving relatively low error, the averaging method is less practical in situations where an understanding of uncertainty or the probabilistic nature of the state estimate is critical.

Figure 3.5(b) illustrates that the trace of the covariance matrix for the WB fusion remains consistently higher than that of the centralized KF. Similar to the Gaussian noise scenario, the WB fusion demonstrates greater uncertainty due to its distribution-averaging approach, whereas the KF follows a recursive process designed to minimize uncertainty via the Kalman Gain. Although the KF achieves a lower covariance trace, its accuracy is compromised by the non-Gaussian nature of the noise, which the WB is better equipped to manage—albeit at the expense of increased uncertainty.

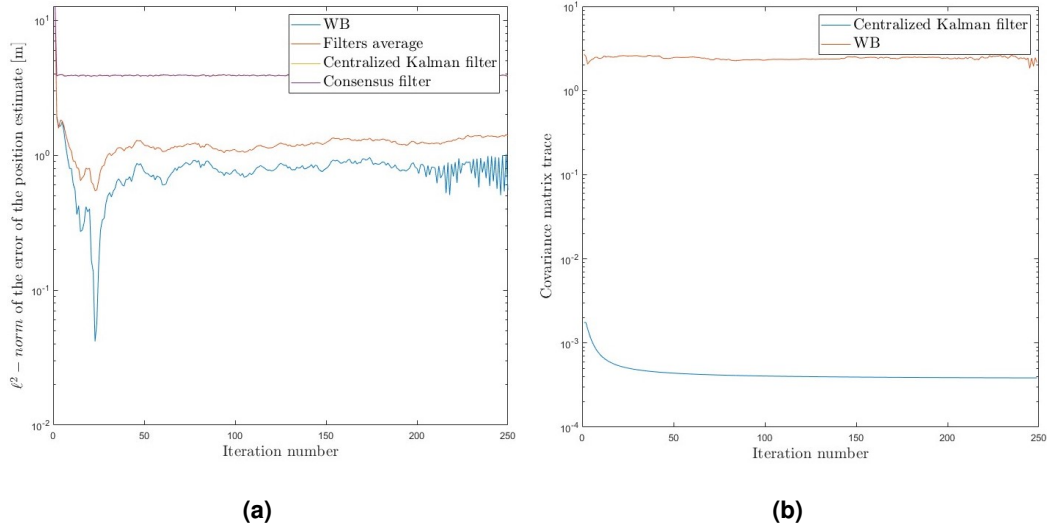
The results in Table 3.2 and Figure 3.5(c) highlight the superior performance and robustness of the WB fusion in scenarios involving heterogeneous noise. Both the table and the figure indicate that the WB fusion consistently achieves lower RMSE compared to the centralized KF, consensus algorithm, and averaging approach. The lower RMSE and reduced variability across simulations suggest that the WB method is well-suited for managing the diverse noise profiles encountered by the sensors.

Notably, the reduced variability of the WB fusion across multiple simulations, as indicated by the standard deviation in Table 3.2, aligns with the stability observed in Figure 3.5(c). The WB consistently performs well despite differing noise realizations, whereas both the consensus algorithm and the centralized KF exhibit larger fluctuations, indicating their sensitivity to noise heterogeneity.

Although the averaging approach yields competitive RMSE values, its inability to convey information regarding the uncertainty or the PDF of the estimate remains a significant limitation.

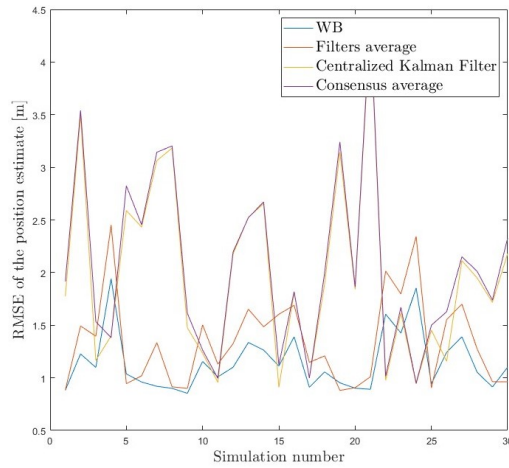
Overall, the WB fusion demonstrates superior adaptability and robustness in environments characterized by heterogeneous noise. While the centralized KF is optimal under Gaussian noise assumptions, it becomes less effective when noise deviates from these assumptions. In contrast, the WB method, with

its general-purpose fusion capability, successfully handles both non-Gaussian and varying noise conditions without requiring specific assumptions about the noise distribution. This versatility makes the WB a powerful tool for sensor fusion in complex and dynamic environments.



(a)

(b)



(c)

**Figure 3.5:** Different noises simulation: a)  $\ell^2$ -norm of the position error; b) Covariance matrix trace of the position uncertainty; c) RMSE over multiple simulations.

|                             | WB     | Filters average | Centralized KF | Consensus average |
|-----------------------------|--------|-----------------|----------------|-------------------|
| Mean RMSE [m]               | 1.1473 | 1.3462          | 1.9733         | 2.0569            |
| RMSE Standard Deviation [m] | 0.2802 | 0.4304          | 0.8658         | 0.8505            |

**Table 3.2:** RMSE of different noises simulation.

# 4

## Conclusion

### Contents

---

|                           |    |
|---------------------------|----|
| 4.1 Summary . . . . .     | 61 |
| 4.2 Future Work . . . . . | 61 |

---

## 4.1 Summary

This thesis has examined state estimation (SE) in linear systems, with particular emphasis on addressing both Gaussian and non-Gaussian noise assumptions within single-agent and multi-agent architectures. The primary objective of this work was to enhance estimation accuracy and robustness, especially in scenarios where traditional approaches, such as the KF, prove inadequate due to their reliance on Gaussian noise assumptions. To this end, we introduced novel methodologies leveraging advanced filtering techniques, including the CF filter, to extend the capabilities of state-space models for managing more complex noise distributions.

For single-agent systems, we revisited foundational filtering techniques such as the KF and the PF, highlighting their inherent limitations in addressing non-Gaussian noise. To overcome these limitations, we developed and evaluated an efficient hybrid filtering approach that combines the strengths of Bayesian frameworks with characteristic function filtering, thus optimizing performance under non-Gaussian noise conditions. Our results demonstrated that the hybrid filter substantially outperforms traditional methods in terms of accuracy, particularly in scenarios involving non-Gaussian disturbances.

For multi-agent architectures, we investigated distributed SE and proposed sensor fusion techniques based on the WB. These methods showed significant promise in settings where sensors experienced heterogeneous noise characteristics. The WB-based fusion approach was found to enhance both accuracy and robustness in multi-sensor environments, providing reliable state estimates despite non-Gaussian noise and sensor diversity.

Extensive simulations and experimental evaluations were conducted to validate the proposed methods. The results revealed substantial improvements in estimation accuracy across a range of noise conditions and sensor architectures. The proposed approaches not only outperformed traditional SE techniques but also demonstrated practical applicability, thereby contributing valuable advancements to the field of state estimation.

## 4.2 Future Work

While the contributions of this thesis have addressed several critical aspects of SE in linear systems, there remain numerous open challenges and promising directions for further research. Specifically, the following areas could be pursued to extend and refine the methodologies proposed:

- **Refinement of the Hybrid Filter via Gaussian Mixture Approximation:** A crucial component of the Hybrid Filter presented in this thesis is the optimization involved in the Gaussian Mixture Approximation. Future research could focus on refining this optimization process to further enhance both accuracy and computational efficiency. In particular, developing more sophisticated

algorithms to improve the approximation of characteristic functions while managing the complexity of the mixture components represents a promising direction for advancing the hybrid filter's performance.

- **Testing the WB Fusion Algorithm in Distributed Settings:** The WB-based sensor fusion algorithm developed in this thesis has demonstrated significant potential for enhancing estimation accuracy in multi-agent systems. However, its implementation thus far has been limited to centralized settings. An important next step would be to implement and evaluate the WB fusion algorithm in a distributed sensor network. Such a distributed implementation would enable more robust and scalable SE, particularly in scenarios where sensors are geographically dispersed and centralized processing is impractical. Key challenges in this direction include ensuring consensus among distributed agents, mitigating communication delays, and overcoming other inherent difficulties in decentralized networks.

# Bibliography

- [1] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," vol. 82, pp. 35–45, 1960.
- [2] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," I. Kadar, Ed., Orlando, FL, USA, Jul. 1997, p. 182. [Online]. Available: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.280797>
- [3] E. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, Oct. 2000, pp. 153–158.
- [4] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002, conference Name: IEEE Transactions on Signal Processing. [Online]. Available: <https://ieeexplore.ieee.org/document/978374>
- [5] M. Idan and J. L. Speyer, "An estimation approach for linear stochastic systems based on characteristic functions," *Automatica*, vol. 78, pp. 153–162, Apr. 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0005109816305465>
- [6] A. P. Vinod, B. Homchaudhuri, and M. M. K. Oishi, "Forward Stochastic Reachability Analysis for Uncontrolled Linear Systems using Fourier Transforms," in *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*, Apr. 2017, pp. 35–44, arXiv:1610.04550 [cs, math]. [Online]. Available: <http://arxiv.org/abs/1610.04550>
- [7] H. Zhang, "Non-linear Bayesian filtering by convolution method using fast Fourier transform," *Fusion*, pp. 1–6, Jul. 2011, mAG ID: 2162005837 S2ID: 23ef6b27aebcee4dceb95434e8ace7d7cbd85b72.
- [8] A. G. Wills, J. Hendriks, C. Renton, and B. Ninness, "A Bayesian Filtering Algorithm for Gaussian Mixture Models," Jun. 2023, arXiv:1705.05495 [cs, stat]. [Online]. Available: <http://arxiv.org/abs/1705.05495>

- [9] S. He, M.-J. Tahk, H.-S. Shin, S. Xu, and A. Tsourdos, "Distributed estimation over a low-cost sensor network: a review of state-of-the-art," *Information Fusion*, vol. 54, pp. 21–43, Feb. 2020, mAG ID: 2951450390 S2ID: 5bf57ad18f25bfb2733e77eed89628de539e253e.
- [10] R. Olfati-Saber and J. Shamma, "Consensus Filters for Sensor Networks and Distributed Sensor Fusion," in *Proceedings of the 44th IEEE Conference on Decision and Control*, Dec. 2005, pp. 6698–6703, iISSN: 0191-2216.
- [11] R. Olfati-Saber, "Distributed Kalman Filter with Embedded Consensus Filters," in *Proceedings of the 44th IEEE Conference on Decision and Control*, Dec. 2005, pp. 8179–8184, iISSN: 0191-2216.
- [12] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, Apr. 2005, pp. 63–70.
- [13] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *2007 46th IEEE Conference on Decision and Control*, Dec. 2007, pp. 5492–5498, iISSN: 0191-2216.
- [14] K. Shen, Z. Jing, and P. Dong, "A Consensus Nonlinear Filter With Measurement Uncertainty in Distributed Sensor Networks," *IEEE Signal Processing Letters*, vol. 24, no. 11, pp. 1631–1635, Nov. 2017, conference Name: IEEE Signal Processing Letters.
- [15] J. Li, F. Deng, and J. Chen, "A Fast Distributed Variational Bayesian Filtering for Multisensor LTV System With Non-Gaussian Noise," *IEEE Transactions on Cybernetics*, vol. 49, no. 7, pp. 2431–2443, Jul. 2019, conference Name: IEEE Transactions on Cybernetics.
- [16] G. Battistelli and L. Chisci, "Kullback–Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability," *Automatica*, vol. 50, no. 3, pp. 707–718, Mar. 2014. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0005109813005591>
- [17] A. Runnalls, "Kullback-Leibler Approach to Gaussian Mixture Reduction," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 3, pp. 989–999, Jul. 2007, conference Name: IEEE Transactions on Aerospace and Electronic Systems. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/4383588>
- [18] C. A. Uribe, D. Dvinskikh, P. Dvurechensky, A. Gasnikov, and A. Nedić, "Distributed Computation of Wasserstein Barycenters over Networks," Sep. 2018, arXiv:1803.02933 [cs, math, stat]. [Online]. Available: <http://arxiv.org/abs/1803.02933>