NEW METHODS FOR DECENTRALISED SENSOR FUSION AND EXTENDED TARGET TRACKING MODELS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HİLAL KÖKSAL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

DECEMBER 2022

Approval of the thesis:

NEW METHODS FOR DECENTRALISED SENSOR FUSION AND EXTENDED TARGET TRACKING MODELS

submitted by **HİLAL KÖKSAL** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İlkay Ulusoy
Head of Department, **Electrical and Electronics Engineering** _____

Assoc. Prof. Dr. Emre Özkan
Supervisor, **Electrical and Electronics Engineering, METU** _____

Examining Committee Members:

Assoc. Prof. Dr. Elif Vural
Electrical and Electronics Engineering, METU _____

Assoc. Prof. Dr. Emre Özkan
Electrical and Electronics Engineering, METU _____

Prof. Dr. Umut Orguner
Electrical and Electronics Engineering, METU _____

Assoc. Prof. Dr. Mustafa Mert Ankaralı
Electrical and Electronics Engineering, METU _____

Assoc. Prof. Dr. Cem Tekin
Electrical and Electronics Engineering, Bilkent University _____

Date: 16.12.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Hilal Köksal

Signature :

ABSTRACT

NEW METHODS FOR DECENTRALISED SENSOR FUSION AND EXTENDED TARGET TRACKING MODELS

Köksal, Hilal
M.S., Department of Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. Emre Özkan

December 2022, 111 pages

The focus of this thesis is two-fold; the first study investigates the use of student-t distribution for the decentralised multi-sensor fusion problem. Multi-sensor fusion can suffer from several artefacts such as low channel capacity, delays in the communication channels, the correlation in the acquired information and sensor biases. When combined with these artefacts, the errors in the local sensors' estimates can result in conflicting information at the fusion center. Such conflicts may later be resolved by future observations. Traditional Gaussian fusion can perform poorly in such cases due to its inherent uni-modal assumption. We propose incorporating student-t distribution while performing multi-sensor fusion, which introduces the ability to represent the uncertainty due to conflicting sensor information. Another focus of this thesis is an alternative measurement update framework for Gaussian process-based extended target tracking models. The proposed method performs variational inference in the measurement update step to improve the accuracy of the kinematic and extent state estimates. The performance evaluations of the methods are presented by conducting various simulations and real data experiments.

Keywords: multi sensor fusion, student-t distribution, extended target tracking, Gaussian process, variational Bayes

ÖZ

MERKEZİ OLMAYAN SENSÖR FÜZYONU VE GENİŞLETİLMİŞ HEDEF TAKİBİ MODELLERİ İÇİN YENİ YÖNTEMLER

Köksal, Hilal
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Emre Özkan

Aralık 2022 , 111 sayfa

Bu tezin odak noktası iki türlü olmakla beraber ilk çalışma, merkezi olmayan çoklu sensör füzyon problemi için student-t dağılımının kullanımına yoğunlaşmaktadır. Çoklu sensör füzyonunun performansında, sınırlı kapasite, iletişim kanallarındaki gecikmeler, elde edilen bilgilerdeki korelasyon veya sensörlerdeki ölçüm hataları gibi sebeplerden dolayı bir kayıp gerçekleşebilir. Tüm bu sebepler gözetildiğinde, yerel sensörlerin tahminlerindeki hatalar füzyon merkezinde çelişkili bilgilerle sonuçlanabilir. Öte yandan, çelişen bilginin sonradan gelen doğru bilgi ile düzeltilebilmesi mümkün olabilir. Bu bağlamda, geleneksel Gauss füzyonu, tek tepecikli yapısı nedeniyle yanıltıcı sonuçlar doğurabilir. Bu durumun sonucunda, student-t dağılımı çoklu sensör füzyona adapte edilerek çelişen bilgi durumunda belirsizliğin korunması hedeflenmektedir. Bu tez kapsamında odaklanılan bir diğer çalışma Gauss süreci temelli genişletilmiş hedef takibi yaklaşımına farklı bir ölçüm güncelleme adımı ile alternatif oluşturabilecek bir yaklaşımı ele almaktadır. Önerilen yöntem, kinematik ve şekilsel karakteristik tahminlerinin doğruluğunu iyileştirmek için ölçüm güncelleme adımında varyasyon çıkarımı gerçekleştirmektedir. Belirtilen yaklaşımların performans-

larının değerlendirilmesi konusunda simülasyon ve gerçek verilerden oluşan testlere başvurulmaktadır.

Anahtar Kelimeler: çoklu sensör füzyonu, student t-dağılımı, genişletilmiş hedef takibi, Gauss süreci, varyasyonal Bayes

To my mother

ACKNOWLEDGMENTS

First of all, I would like to thank my supervisor Emre Özkan for his enlightening remarks and sage pieces of advice. His constant support, guidance, and patience made this voyage much smoother.

Being a part of Sensor Fusion Laboratory with great members was a pleasure. Specifically, I must mention Murat Kumru for his unwavering support and friendship.

I have had the good fortune to work at Kuartis for the last year of my MSc in an environment surrounded by nothing but kindness and full of understanding. All my colleagues are appreciated, but I must single out Berker Loğoğlu, Oğuz Özdemir, Eren Şener, Elvan Ülker and Uğur Uyanık for their constant support and friendship.

I thank TÜBİTAK, The National Scientific and Technological Reseach Council of Turkey for supporting me with 2210A M.S Studies scholarship.

The support and respite provided by my friends have been vital as I have tried to navigate all the highs and lows of the MSc journey. I would like to thank and acknowledge Melike Kaya, Hatice Elif Tola and Tuğçe Deniz for their love and encouragement.

I express my sincere gratitude to my loving parents, İmran and Cahit Köksal, who have always helped me in whatever way possible. And, of course, I would like to offer my special thanks to my brother Kaan Köksal for his constant motivation throughout my research work.

This work was financially supported by Turkcell Technology A.Ş., under the '5G and Beyond Graduate Support Program', organized by the Information and Communication Technologies Authority of the Republic of Turkey.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

2D	2 Dimensional
3D	3 Dimensional
BC	Bar-Shalom/Campo State Vector Combination
BLUE	Best Linear Unbiased Estimation
CC	Complex Combination
CFST	Chernoff Fusion with Student t-Distribution
CI	Covariance Intersection
CKF	Centralized Kalman Filter
EKF	Extended Kalman Filter
GMM	Gaussian Mixture Models
GP	Gaussian Process
HOT	Higher Order Terms
IPM	Inverse Perspective Mapping
KF	Kalman Filter
KL	Kullback-Leibler
LiDAR	Light Detection and Ranging
IOU	Intersection over Union
PDF	Probability Distribution Function
RADAR	Radio Detection and Ranging
RMSE	Root-Mean-Square Error
UKF	Unscented Kalman Filter

CHAPTER 1

INTRODUCTION

Multi-sensor fusion is a strategy of merging measurements from various sensors to create a solid and comprehensive description of surroundings or processes of interest. The notion of multi-sensor fusion is based on the fact that combining complementary data from several sensors, which results in better accuracy of solutions to information processing issues. This concept is currently being applied in many areas and continues to gain ground in new ones. As an example, many robotics fields, including localization, and environment mapping, find extensive use for multi-sensor data fusion.

Due to the strengths and weaknesses of each sensor, sensor fusion is required to merge information from individual sensors into a single shot. As an advantage, the processing and interpretation of the information become more efficient thanks to combining information from multiple sensors, which results in decision-making more accurately and quickly. Furthermore, multi-sensor fusion enables us to compensate for the errors inherent in individual sensors. For instance, if one sensor reports erroneous information, the other can provide supplementary data to fix this error.

The main concern of this study in terms of multi-sensor fusion is related to a specific fusion algorithm, *so called* Chernoff fusion. A specific and standard version of the Chernoff fusion is constructed with the Gaussian density assumption as in [1, 2, 3] since a closed-form solution is available. The Chernoff fusion can also be established using student-t distributions rather than the Gaussian assumption. In addition to the particular characteristics similar to Gaussian distribution, insufficient or misleading data makes student-t distribution a better option to deal with uncertainty. The reason is that student-t accommodates the occurrence of the values with higher dispersion.

The greater variability is represented in the distribution with the characteristics such as lower peak and heavier tails than the Gaussian distribution. The introduction of heavy-tailed posteriors using student-t distributions allows us to maintain the multiple hypotheses about the unknown variable as long as possible for the Chernoff fusion. In this study, we consider the problem of fusing inconsistent sensor information where the uncertainty must be preserved until it is resolved by new information. A traditional approach which relies on Gaussian distributions can fail to do so, as the information provided by the sensors is summarized by another Gaussian. On the other hand, a fusion algorithm constructed with student-t distribution enables us to deal with the performance degradation due to the conflicting sensor information.

Another focus of this thesis is related to the extended object tracking framework. Some applications, such as autonomous driving, render the point object assumption invalid due to objects in the sensors' near field. Thanks to the rapid advances in the capability of the sensors, the state definition, which involves kinematics, is expanded with the object's extent. Although various formulations are available to estimate the object's extent in addition to the kinematics, this study concentrates on a specific approach constructed with the Gaussian process. In early work, [4], Gaussian Process is employed to learn and track the target's shape by deriving a state space model to be used with the Kalman filter. [4] updates extent using predicted state of the kinematics. This results in dependency on the compatibility of the process model with the situation or dependency on the requirement of a small sampling time to have reasonable extent estimates. This work investigates an alternative measurement update framework for extended object tracking that identifies the extent using the Gaussian process. Furthermore, the new measurement update framework is built on the variational Bayes. Variational Bayes, the inference technique to obtain the state vector, aims to minimize the difference between probability distributions using a concept for measuring the relative distance in an iterative procedure. Based on this definition, the variational inference method enables us to use updated estimates for kinematics rather than predicted ones to update the extent. Further, updating kinematics and extent in a given order at each time is repeated a predefined number of iteration times. Improvement in the estimate of extent promotes a better estimate of kinematics. As a result of enhancing each other, this iteration process makes approximate solutions for

kinematics and extent closer to their optimal solutions.

This thesis is composed of five chapters. Chapter 1 briefly explains the focus and the motivation for this study. The rest of the thesis is organized as follows: Chapter 2 gives theoretical background on the principles that serve as the foundation for the thesis's two primary subjects: Bayesian filtering, student's t-distribution, Chernoff fusion, Gaussian process, and variational inference. Chapter 3 provides the literature review and a toy example to clarify the motivation for the Chernoff fusion with student t-distribution. Further, the algorithm's derivation is given in steps that construct the building blocks of the method, and the performance evaluation is achieved through simulations and real data experiments. Chapter 4 presents the literature review and problem definition of the variational measurement update framework for extended object tracking using Gaussian process. Moreover, the measurement update equations are provided together with the simulations and real data experiments for performance evaluation. Finally, Chapter 5 outlines a brief conclusion.

CHAPTER 2

THEORETICAL BACKGROUND

2.1 Bayesian Filtering

Some essential engineering applications such as navigation, target tracking, telecommunications, signal processing, and control require an effective way to acquire the necessary information using noisy measurements. Based on the application, the state's definition, which consists of different dynamic variables like position, velocity, and orientation, encapsulates all the critical information to describe the system. The main problem of optimal or Bayesian filtering is estimating the dynamic state variables directly or indirectly through noisy measurements, which introduce a spread of possible values for the state elements [5]. Furthermore, another source of ambiguity in estimating the state variables is the process noise to model the uncertainty of the dynamic model, which represents the state's evolution over time. The state-space model, which describes both the dynamic behavior of the state and the relation between state and measurement, can be written as follows.

$$x_k = f(x_{k-1}, w_{k-1}), \tag{2.1a}$$

$$z_k = h(x_k, v_k), \tag{2.1b}$$

where x_k and z_k stand for state and noisy measurement at time k, respectively. Dynamic behavior of the state is modeled by the function *f(.)* whose inputs are the previous state x_{k-1} and white process noise w_{k-1}. The function *h(.)* constructs the relation between state and measurements with the white measurement noise v_k as input. The measurement model *h(.)* differs based on the sensor type and state definition. Furthermore, x_0, w_k and v_k are independent from each other for each instant k.

From the Bayesian framework, estimation of state variables through noisy measure-

ments turns into the computation of the posterior density function of the state $p(x_k|z_{0:k})$, which makes the probabilistic state-space model more suitable. The following conditional distributions represent the state-space model 2.1 in a probabilistic way

$$x_k \sim p(x_k|x_{k-1}, z_{1:k-1}), \tag{2.2a}$$

$$z_k \sim p(z_k|x_k, z_{1:k-1}), \tag{2.2b}$$

where

- $p(x_k|x_{k-1}, z_{1:k-1})$, transition density, denotes the dynamic behavior of the state x_k. The Markov property assumption introduces the independence of the current state x_k from anything before time step $k-1$, which can be expressed as follows
$$p(x_k|x_{0:k-1}, z_{1:k-1}) = p(x_k|x_{k-1}). \tag{2.3}$$
- $p(z_k|x_k, z_{1:k-1})$, measurement likelihood, implies the measurement model of the system. The independence of current measurement z_k from previous measurements $z_{1:k-1}$ and states $x_{0:k-1}$ is also valid in this case due to Markov model assumption
$$p(z_k|x_{0:k}, z_{1:k-1}) = p(z_k|x_k). \tag{2.4}$$

Given the probabilistic state-space model, the Bayesian filtering recursion is constructed with two steps, called prediction and update. The prediction step aims to calculate the predicted state density $p(x_k|z_{1:k-1})$ based on the Chapman-Kolmogorov equation as follows

$$p(x_k|z_{1:k-1}) = \int p(x_k, x_{k-1}|z_{1:k-1})\, dx_{k-1}, \tag{2.5a}$$

$$= \int p(x_k|x_{k-1}, z_{1:k-1}) p(x_{k-1}|z_{1:k-1})\, dx_{k-1}. \tag{2.5b}$$

The Chapman-Kolmogorov equation 2.5 can be simplified in the case of the Markov model 2.3 as follows

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1}) p(x_{k-1}|z_{1:k-1})\, dx_{k-1}. \tag{2.6}$$

The update step involves the new measurement at time step k to the predicted state density $p(x_k|z_{1:k-1})$. The updated state density or the posterior state density $p(x_k|z_{1:k})$

indicates the outcome of the combination based on Bayes' rule

$$p(x_k|z_{1:k}) = \frac{p(x_k, z_k, z_{1:k-1})}{p(z_k, z_{1:k-1})}, \tag{2.7a}$$

$$= \frac{p(z_k|x_k, z_{1:k-1})p(x_k, z_{1:k-1})}{p(z_k, z_{1:k-1})}, \tag{2.7b}$$

$$= \frac{p(z_k|x_k, z_{1:k-1})p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})}. \tag{2.7c}$$

Further simplification of the update equation 2.7c can be achieved thanks to the assumption 2.4 as follows

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})}. \tag{2.8}$$

The predicted likelihood $p(z_k|z_{1:k-1})$, which corresponds to the normalization constant, enables the following adjustment in the update equation

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{\int p(z_k|x_k)p(x_k|z_{1:k-1})\, dx_k}. \tag{2.9}$$

As can be seen, the outputs of the prediction and the update steps are the inputs of each other, which constructs a recursion starting from the prior distribution $p(x_0)$. Although a closed-form solution is not available for Bayesian filtering in general, a critical exception is described by the Kalman filter.

2.1.1 Kalman Filter

The Kalman filter [6] is a special case where the system dynamics and measurement models 2.1 are linear with additive Gaussian noise, and the initial prior is also Gaussian. The following state-space model satisfies the conditions for a closed-form solution provided by the Kalman filter

$$x_k = A_{k-1}x_{k-1} + w_{k-1}, \tag{2.10a}$$

$$z_k = H_k x_k + v_k, \tag{2.10b}$$

where A_{k-1} is the transition matrix which defines dynamic behavior of x_k, and H_k is the measurement model matrix which constructs a relation between x_k and z_k. $w_{k-1} \sim \mathcal{N}(0, Q_{k-1})$ is the process noise with the process noise covariance matrix Q_{k-1}, and $v_k \sim \mathcal{N}(0, R_k)$ is the measurement noise with the measurement noise covariance matrix R_k.

The probabilistic version of the state-space model 3.44, considering the assumptions 2.3 and 2.4, can be written as

$$p(x_k|x_{k-1}) \sim \mathcal{N}(A_{k-1}x_{k-1}, Q_{k-1}), \tag{2.11a}$$

$$p(z_k|x_k) \sim \mathcal{N}(H_k x_k, R_k), \tag{2.11b}$$

$$p(x_0) \sim \mathcal{N}(x_0, P_0). \tag{2.11c}$$

The linear and Gaussian model also makes the resulting distributions of the Bayesian filtering Gaussian. The continuity of the Gaussian form brings about an advantage related to the sufficiency of mean and covariance to get all the information about the state. As a result, Kalman filter equations are based on the computation of the mean and covariance recursively. The whole procedure can be evaluated in a closed form

$$p(x_k|z_{1:k-1}) \sim \mathcal{N}(x_k; \mu_{k|k-1}, P_{k|k-1}), \tag{2.12a}$$

$$p(x_k|z_{1:k}) \sim \mathcal{N}(x_k; \mu_{k|k}, P_{k|k}), \tag{2.12b}$$

$$p(z_k|z_{1:k-1}) \sim \mathcal{N}(z_k; H_k\mu_{k|k-1}, S_k), \tag{2.12c}$$

where the sufficient statistics of the distributions are evaluated as

- Prediction step:

$$\mu_{k|k-1} = A_{k-1}\mu_{k-1|k-1}, \tag{2.13a}$$

$$P_{k|k-1} = A_{k-1}P_{k-1|k-1}A_{k-1}^\top + Q_{k-1}. \tag{2.13b}$$

- Update step:

$$S_k = H_k P_{k|k-1} H_k^\top + R_k, \tag{2.14a}$$

$$K_k = P_{k|k-1} H_k^\top S_k^{-1}, \tag{2.14b}$$

$$\mu_{k|k} = \mu_{k|k-1} + K_k(z_k - H_k\mu_{k|k-1}), \tag{2.14c}$$

$$P_{k|k} = P_{k|k-1} - K_k S_k K_k^\top, \tag{2.14d}$$

where μ and P stand for mean and covariance, respectively. Although the Kalman filter is the minimum mean squared error estimator for the linear Gaussian systems, many real-life situations involving nonlinear motion and measurement models can

not be justified appropriately by the Kalman filter assumptions. Since the output of a nonlinear function with a Gaussian input is not Gaussian anymore, an approximation of the non-Gaussian filtering distributions as Gaussian is a solution for the case with nonlinear models.

2.1.2 Extended Kalman Filter (EKF)

The nonlinear characteristics of the many phenomena in nature reveal the requirement of the nonlinear models, which are incompatible with the Kalman filter. On the other hand, the Taylor series expansion-based extended Kalman filter handles such situations under the linearization of the nonlinear equations around the best available information.

The nonlinear Gaussian state-space model with the additive process and measurement noise assumption can be written as follows.

$$x_k = f(x_{k-1}) + w_k, \tag{2.15a}$$

$$z_k = h(x_k) + v_k, \tag{2.15b}$$

where *f(.)* and *h(.)* are the nonlinear dynamic and measurement model functions, whose inputs or outputs are the state x_k and the measurement z_k. $w_{k-1} \sim \mathcal{N}(0, Q_{k-1})$ and $v_k \sim \mathcal{N}(0, R_k)$ stand for process and measurement noise, respectively. Expansion of $f(x_{k-1})$ in the Taylor series about the best available information $\mu_{k-1|k-1}$ enables local linearization, which works well in practice for moderate nonlinearities. The equivalent action for $h(x_k)$ is performed about $\mu_{k|k-1}$

$$f(x_{k-1}) = f(\mu_{k-1|k-1}) + \frac{\partial f}{\partial x}\Big|_{x=\mu_{k-1|k-1}} (x_{k-1} - \mu_{k-1|k-1}) + HOT, \tag{2.16a}$$

$$h(x_k) = h(\mu_{k|k-1}) + \frac{\partial h}{\partial x}\Big|_{x=\mu_{k|k-1}} (x_k - \mu_{k|k-1}) + HOT, \tag{2.16b}$$

where HOT stands for higher order terms. If higher order terms are omitted, the state-space model 2.15 can be approximated as

$$x_k \approx f(\mu_{k-1|k-1}) - F_{k-1}\mu_{k-1|k-1} + F_{k-1}x_{k-1} + w_k, \tag{2.17a}$$

$$z_k \approx h(\mu_{k|k-1}) - H_k\mu_{k|k-1} + H_k x_k + v_k, \tag{2.17b}$$

where the Jacobians can be written as:

$$F_{k-1} = \frac{\partial f}{\partial x}\Big|_{x=\mu_{k-1|k-1}}, \tag{2.18a}$$

$$H_k = \frac{\partial h}{\partial x}\Big|_{x=\mu_{k|k-1}}. \tag{2.18b}$$

The approximated state-space model 2.17 meets the requirement of the Kalman filter related to linearity, which brings the following EKF equations constructed with Jacobians.

- Prediction step:

$$\mu_{k|k-1} = f(\mu_{k-1|k-1}), \tag{2.19a}$$

$$P_{k|k-1} = F_{k-1}P_{k-1|k-1}F_{k-1}^{\top} + Q_{k-1}. \tag{2.19b}$$

- Update step:

$$S_k = H_k P_{k|k-1} H_k^{\top} + R_k, \tag{2.20a}$$

$$K_k = P_{k|k-1} H_k^{\top} S_k^{-1}, \tag{2.20b}$$

$$\mu_{k|k} = \mu_{k|k-1} + K_k(z_k - h(\mu_{k|k-1})), \tag{2.20c}$$

$$P_{k|k} = P_{k|k-1} - K_k S_k K_k^{\top}. \tag{2.20d}$$

An extended Kalman filter, a frequently used method in the nonlinear state-space models, brings with it some restrictions to ensure adequate performance, such as the need for differentiable models. Furthermore, the performance is strictly related to the superiority of the first two terms to the remaining ones in the 2.16. As the higher-order terms become more dominant, the performance degrades more. Additionally, computational complexity is another restriction aspect due to calculating Jacobian matrices with numeric methods when the analytical analysis is not feasible.

2.1.3 Unscented Transform

Unscented transform is based on changing the order of operations to determine the statistics of a random variable subjected to a nonlinear transformation [7]. Rather

than approximating arbitrary nonlinear transformation of a probability distribution, applying the approximation to the probability distribution first and then nonlinear transformation can be interpreted as the idea of the unscented transform. The relation between two random variables through a nonlinear transformation can be expressed as

$$y = f(x) \tag{2.21}$$

where $x \in \mathbb{R}^m$ is a random vector with mean $\bar{x}$ and covariance Σ_x. The function *f(.)* represents the nonlinear transformation. Another random variable y has mean $\bar{y}$ and covariance Σ_y, and also denotes the nonlinear transformation of the random variable x. The approximation of the probability distribution is achieved through sigma points computed in a deterministic way as

$$\begin{aligned} s^j &\triangleq \begin{cases} \bar{x}, & j = 1 \\ \bar{x} + \left[\sqrt{\frac{m}{1-W^0}\Sigma_x}\right]_{:,j-1}, & 2 \leq j \leq m+1 \\ \bar{x} - \left[\sqrt{\frac{m}{1-W^0}\Sigma_x}\right]_{:,j-m-1}, & m+2 \leq j \leq 2m+1 \end{cases} \\ W^j &\triangleq \begin{cases} W^0, & j = 1 \\ \frac{1-W^0}{2m}, & 2 \leq j \leq 2m+1 \end{cases} \end{aligned} \tag{2.22}$$

where m is dimension of the random variable x, $[\cdot]_{:,j}$ denotes the j^{th} column of the matrix argument, $\sqrt{\cdot}$ indicates the matrix square root operation. Furthermore, weights must satisfy the following condition

$$\sum_{j=1}^{m} W^j = 1. \tag{2.23}$$

Since sigma points except $\bar{x}$ are calculated by taking into account the Σ_x based deviation from $\bar{x}$ in all dimensions, $2m+1$ sigma points are available. The propagation of the each sigma points through nonlinear transformation results in a set of transformed sigma points as given

$$y^j = f(s^j). \tag{2.24}$$

The transformed sigma points are used to find out the statistics of the random variable y as follows

$$\bar{y} = \sum_{j=1}^{m} W^j y^j, \tag{2.25}$$

$$\Sigma_y = \sum_{j=1}^{m} W^j (y^j - \bar{y})(y^j - \bar{y})^\top. \tag{2.26}$$

Unscented transform enables to compute statistics after nonlinear transformation using sigma points rather than linearization around the mean as Taylor expansion does and constructs the basis of the unscented Kalman filter.

2.1.4 Unscented Kalman Filter (UKF)

The unscented Kalman filter, an alternative to the extended Kalman filter to handle the nonlinear characteristics of many phenomena, extends the concept of the unscented transform to the recursive estimation [8]. As a result, some drawbacks of extended Kalman filter, such as the derivation of Jacobians which may lead to implementation difficulties and unstable performance in case of elusive local linearity, are not valid for unscented Kalman filter. Such advantages make the UKF a compelling option compared to the EKF when the Kalman filter is not suitable for the case.

The nonlinear Gaussian state-space model with the additive process and measurement noise assumption as in the extended Kalman filter section is as given

$$x_k = f(x_{k-1}) + w_k, \tag{2.27a}$$

$$z_k = h(x_k) + v_k, \tag{2.27b}$$

where *f(.)* and *h(.)* are the nonlinear dynamic and measurement model functions, whose inputs or outputs are the state $x_k \in \mathbb{R}^m$ and the measurement z_k with process and measurement noises $w_{k-1} \sim \mathcal{N}(0, Q_{k-1})$ and $v_k \sim \mathcal{N}(0, R_k)$, respectively. The statistics of the state distribution at time k-1 are represented by the sigma points and their weights, which enables us to deal with the nonlinear dynamic function *f(.)*

and measurement function *h(.)* as follows

$$\mu^j_{k-1|k-1} \triangleq \begin{cases} \mu_{k-1|k-1}, & j = 1 \\ \mu_{k-1|k-1} + \left[\sqrt{\frac{m}{1-W^0}\Sigma_{k-1|k-1}}\right]_{:,j-1}, & 2 \leq j \leq m+1 \\ \mu_{k-1|k-1} - \left[\sqrt{\frac{m}{1-W^0}\Sigma_{k-1|k-1}}\right]_{:,j-m-1}, & m+2 \leq j \leq 2m+1 \end{cases}$$

$$W^j \triangleq \begin{cases} W^0, & j = 1 \\ \frac{1-W^0}{2m}. & 2 \leq j \leq 2m+1 \end{cases} \tag{2.28}$$

Sigma points and the corresponding weights represent the state distribution, leading to the following UKF equations to deal with the nonlinearity.

- Prediction step:

  The mean and the covariance of the predicted state density are computed using the weighted average and outer product of the sigma points propagated through dynamic model function *f(.)*, respectively.

$$\mu^j_{k|k-1} = f(\mu^j_{k-1|k-1}), \tag{2.29}$$

$$\mu_{k|k-1} = \sum_{j=1}^{m} W^j \mu^j_{k|k-1}, \tag{2.30}$$

$$\Sigma_{k|k-1} = \sum_{j=1}^{m} W^j (\mu^j_{k|k-1} - \mu_{k|k-1})(\mu^j_{k|k-1} - \mu_{k|k-1})^\top + Q_{k-1}. \tag{2.31}$$

- Update step:

  The mean and covariance computation of the posterior state density requires predicted measurement $z_{k|k-1}$ and Kalman gain K_k calculated by the innovation covariance S_k and cross-covariance C^{xz}_k. Transformed sigma points $\mu^j_{k|k-1}$ of the prediction step are the input of the nonlinear measurement function *h(.)* to find out the predicted measurement $z_{k|k-1}$. As in the prediction step, the innovation covariance S_k and the cross-covariance C^{xz}_k are computed with weighted

outer product of the transformed sigma points.

$$z^j_{k|k-1} = h(\mu^j_{k|k-1}), \tag{2.32}$$

$$z_{k|k-1} = \sum_{j=1}^{m} W^j z^j_{k|k-1}, \tag{2.33}$$

$$S_k = \sum_{j=1}^{m} W^j (z^j_{k|k-1} - z_{k|k-1})(z^j_{k|k-1} - z_{k|k-1})^\top + R_k, \tag{2.34}$$

$$C^{xz}_k = \sum_{j=1}^{m} W^j (\mu^j_{k|k-1} - \mu_{k|k-1})(z^j_{k|k-1} - z_{k|k-1})^\top, \tag{2.35}$$

$$K_k = C^{xz}_k S^{-1}_k, \tag{2.36}$$

$$\mu_{k|k} = \mu_{k|k-1} + K_k(z_k - z_{k|k-1}), \tag{2.37}$$

$$P_{k|k} = P_{k|k-1} - K_k S_k K_k^\top. \tag{2.38}$$

2.2 Student's t-Distribution

A student's t-distribution or t-distribution is a probability distribution developed under the concern for the inference through a small sample size with an unknown population variance. The student's t-distribution and Gaussian distribution share specific characteristics regarding the bell-shaped curve and symmetry. Still, the student's t-distribution has heavier tails and a lower peak, which introduces more uncertainty. The similarity and difference between Gaussian and student t-distribution can be constructed by describing the random variables through sampling based approach. The random variable which has a standard normal distribution is given as

$$Z = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{N}}}, \tag{2.39}$$

where N observations $\{X_1, X_2, \cdots, X_N\}$ are drawn from a normally distributed population $\mathcal{N}(\mu, \sigma)$ with the following sample mean and the variance

$$\bar{X} = \frac{1}{N} \sum_{i=1}^{N} X_i, \tag{2.40a}$$

$$S^2 = \frac{1}{N-1} \sum_{i=1}^{N} (X_i - \bar{X})^2. \tag{2.40b}$$

Rather than using the unknown variable σ, the sample standard deviation S, which varies based on the sample, brings in the following random variable, which no longer

has the standard normal distribution [9],

$$t = \frac{\bar{X} - \mu}{\frac{S}{\sqrt{N}}}, \tag{2.41}$$

where the random variable t has the student-t distribution with $N - 1$ degrees of freedom. Replacement of the population standard deviation with the sample standard deviation presents a greater variance, which creates a basis for the heavier tails. If the typical characteristics of the two distributions are required when sufficient data is unavailable, student t-distribution is a more reasonable option to come up with meaningful outcomes. The generalized version of the student t-distribution, also valid for random vectors $x \in \mathbb{R}^n$, is described as

$$t_\nu(x; \mu, \Sigma) \triangleq \frac{\Gamma((\nu + n)/2)}{\Gamma(\nu/2)|\pi\nu\Sigma|^{1/2}} \left(1 + (x - \mu)^\top (\nu\Sigma)^{-1}(x - \mu)\right)^{-(\nu+n)/2}, \tag{2.42}$$

where μ and Σ denote mean and scale matrix of random variable x and ν represents the degrees of freedom. Besides, the covariance of random variable x is defined as $P = \frac{\nu}{\nu-2}\Sigma$ if $\nu > 2$.

Gaussian distribution can be considered a particular case of student t-distribution since t-distribution reduces to Gaussian as degrees of freedom go to infinity as shown in Figure 2.1. As a result, both claim several convenient properties, and the t-distribution interpretation of these common properties can be listed as follows.

- When an affine transformation is applied to a random vector $x \sim t_\nu(x; \mu, \Sigma)$, the form of the distribution remains unaltered with the following statistics

$$y = Ax + b, \tag{2.43}$$

$$p(y) = t_\nu(y; A\mu + b, A\Sigma A^\top). \tag{2.44}$$

- Let $x_1 \in \mathbb{R}^{n_1}$ and $x_2 \in \mathbb{R}^{n_2}$ be the jointly t-distributed two random vectors described as

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim t_\nu \left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right). \tag{2.45}$$

  The two crucial concepts often mentioned with Gaussian distribution, marginalization and conditioning, are also valid for the student t-distribution. The marginalization of x_1 is achieved by applying a linear transformation with $A = \begin{bmatrix} I & 0 \end{bmatrix}$,

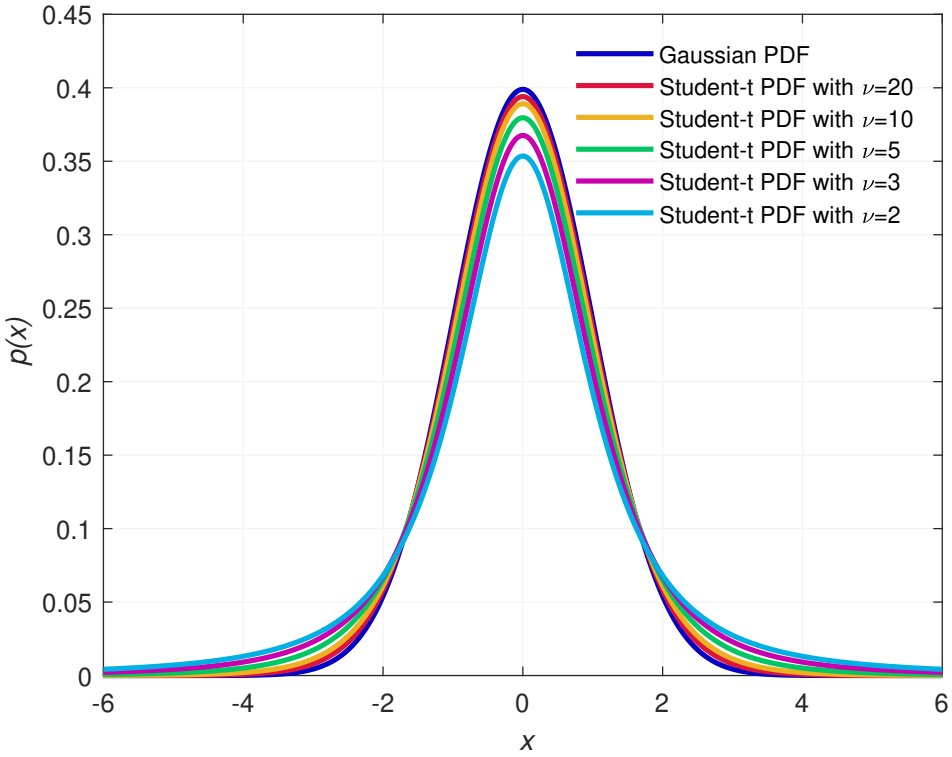


Figure 2.1: PDF of Gaussian distribution and student t-distribution with different degrees of freedom.

which provides the following result

$$p(x_1) = t_\nu(x_1; \mu_1, \Sigma_{11}). \tag{2.46}$$

Further, conditional pdf $p(x_1|x_2)$ is given by the following statistics.

$$p(x_1|x_2) = t_{\nu_{1|2}}(x_1; \mu_{1|2}, \Sigma_{1|2}), \tag{2.47}$$

$$\nu_{1|2} = \nu + n_2, \tag{2.48}$$

$$\mu_{1|2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2), \tag{2.49}$$

$$\Sigma_{1|2} = \frac{\nu + \Delta_2^2}{\nu + n_2}\left(\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{12}^\top\right), \tag{2.50}$$

where $\Delta_2 = (x_2 - \mu_2)^\top(\nu\Sigma_{22})^{-1}(x_2 - \mu_2)$.

2.3 Chernoff Fusion

Fusing information from two different sources is a general scenario to yield an optimal estimation about an unknown variable $x \in \mathbb{R}^n$. The estimates of the two sources are explained in the form of probability density functions $p_{x,1}(\cdot)$ and $p_{x,2}(\cdot)$, which share a piece of common information that comes from past fusion results, communication link failures, process noise, etc. Therefore, identification and removal of common information are of interest. Otherwise, duplicate information may degrade the performance due to overconfidence, divergence and bias in estimates. Mathematically, optimal density fusion can be written as follows

$$p_{x,f}(x) \propto \frac{p_{x,1}(x)p_{x,2}(x)}{p_{x,12}(x)}, \tag{2.51}$$

where $p_{x,12}(\cdot)$ stands for density which represents the common information, and $p_{x,f}(x)$ denotes the optimal fusion density. Although the removal of duplicate information in (2.51) is straightforward in a theoretical sense, some limitations in the practical systems such as communication between sources and computational power encourage using suboptimal methods such as Chernoff fusion based on the following

$$p_{x,cf}(x) = \frac{p_{x,1}^{w}(x)p_{x,2}^{1-w}(x)}{\int p_{x,1}^{w}(x)p_{x,2}^{1-w}(x)\mathrm{d}x}, \tag{2.52}$$

where $p_{x,cf}(x)$ denotes the fused density, and $w \in [0,1]$ is determined using optimization techniques.

The approach in (2.52) is proposed in [10] and [11] as a generalization of CI technique. CI assumes that states are described with Gaussian density function, which also results in fused density as Gaussian. Since a closed-form solution is available for the exponential family of distributions, the Gaussian assumption introduces a closed-form solution. On the other hand, the form of the two arbitrary density functions may not remain the same after the Chernoff fusion. Furthermore, an analytical solution is not accessible for any arbitrary distributions. Assuming $p_{x,1}(\cdot)$ and $p_{x,2}(\cdot)$ are Gaussian

$$p_{x,1}(x) \triangleq \mathcal{N}(x; x_1, P_1), \tag{2.53a}$$

$$p_{x,2}(x) \triangleq \mathcal{N}(x; x_2, P_2). \tag{2.53b}$$

Log-linear combination of $p_{x,1}(x)$ and $p_{x,2}(x)$ followed by normalization as in (2.52) makes Chernoff fusion equivalent to CI technique. The mean x_{CI} and the covariance P_{CI} which describe the fused density are determined as follows

$$P_{\text{CI}}^{-1} x_{\text{CI}} = w P_1^{-1} x_1 + (1-w) P_2^{-1} x_2, \tag{2.54a}$$

$$P_{\text{CI}}^{-1} = w P_1^{-1} + (1-w) P_2^{-1}. \tag{2.54b}$$

One of the approaches to determine $w \in [0, 1]$ in (2.52) is to use maximization/minimization of a cost function as mentioned in [12]. The determinant/trace of the covariance matrix and Shannon/Renyi entropy can be considered as the possible cost functions. An alternative as stated in [12] is to use divergence metrics such as Kullback-Leibler divergence for the measure of relative distance between two distributions.

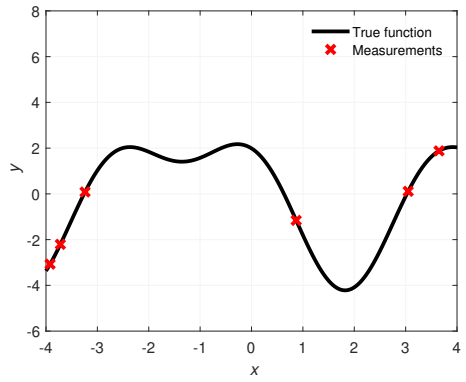
2.4 Gaussian Processes

The Gaussian process, a non-parametric stochastic process, provides a convenient way to obtain a posterior based on distribution over function using a given data set and a prior which carries information about a random function. Since its description is based on probabilistic structure, the Gaussian process represents uncertainty related to the data set and investigated parts of the function. An example function which relates state and the measurement can be written as

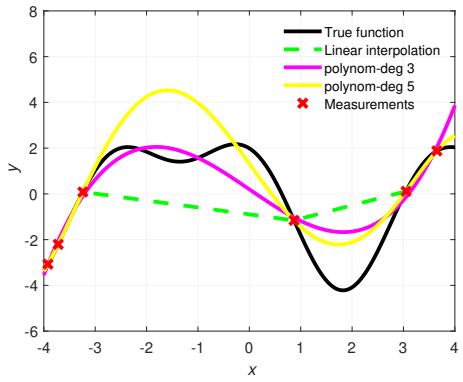
$$z_k = f(x_k) + e_k, \;\; e_k \sim \mathcal{N}(0, \; R) \tag{2.55}$$

where z_k is noisy observation at time k, x_k is the corresponding input to that observation, *f(.)* is the function desired to be known and e_k is the measurement noise. The aim is to learn the function $\mathbf{f} = \begin{bmatrix} f(\bar{x}_1) & \cdots & f(\bar{x}_n) \end{bmatrix}^\top$ at inputs $\bar{\mathbf{x}} = \begin{bmatrix} \bar{x}_1 & \cdots & \bar{x}_n \end{bmatrix}^\top$ by using a set of noisy measurements $\mathbf{z} = \begin{bmatrix} z_1 & \cdots & z_m \end{bmatrix}^\top$ with their inputs $\mathbf{x} = \begin{bmatrix} x_1 & \cdots & x_m \end{bmatrix}^\top$. The Gaussian process is based on the assumption that $\mathbf{z}$ and $\mathbf{f}$ are jointly Gaussian distributed

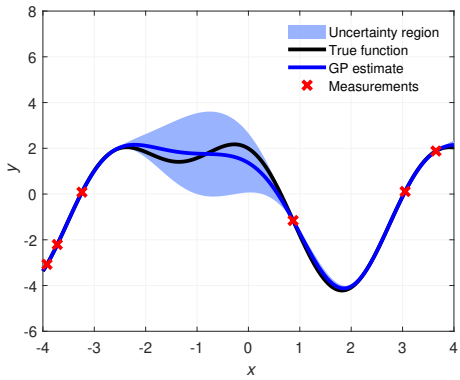
$$\begin{bmatrix} \mathbf{z} \\ \mathbf{f} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(\mathbf{x}, \mathbf{x}) + I_m \otimes R & K(\mathbf{x}, \bar{\mathbf{x}}) \\ K(\bar{\mathbf{x}}, \mathbf{x}) & K(\bar{\mathbf{x}}, \bar{\mathbf{x}}) \end{bmatrix}\right), \tag{2.56}$$



(a)



(b)



(c)

Figure 2.2: (a) displays a set of measurements and the true function. (b) shows functions obtained by linear interpolation and fitting polynomials with degrees 3 and 5 in addition to the true function and the measurements. (c) illustrates the estimate of the function values with uncertainty region using the Gaussian process, the true function, and the measurements.

where

$$K(\mathbf{x}, \bar{\mathbf{x}}) = \begin{bmatrix} k(x_1, \bar{x}_1) & \dots & k(x_1, \bar{x}_n) \\ \vdots & & \vdots \\ k(x_m, \bar{x}_n) & \dots & k(x_m, \bar{x}_n) \end{bmatrix}, \tag{2.57}$$

and $k(\cdot, \cdot)$ is the kernel or the covariance function which determines the degree of correlation between its inputs. The kernel, which captures the assumptions made about the function, has a decisive role in the performance of the Gaussian process. Although several alternatives are available, one popular choice is the squared exponential kernel

function

$$k\left(u,v\right)=\sigma^{2}e^{-\frac{|u-v|^{2}}{2l^{2}}}, \tag{2.58}$$

where σ and l are the variance and the length scale, respectively. Moreover, the squared exponential kernel is a function of $|u-v|$ since the correlation between two inputs decreases as the distance between them grows, which is quite intuitive. Another popular alternative is a modified version of the squared exponential kernel in terms of periodicity

$$k(u,v)=\sigma_{f}^{2}e^{-\frac{2\sin^{2}\left(\frac{|u-v|}{2}\right)}{l^{2}}}, \tag{2.59}$$

where σ_f^2 and l stand for prior variance of signal amplitude and length scale of the function, respectively.

The estimation of the unknown function values is constructed based on conditioning, which is available in a closed form solution due to the nature of the Gaussian process. Two jointly Gaussian random vectors establish that conditional distribution for one vector given the other is also Gaussian as given

$$p(\mathbf{f}|\mathbf{z})\sim\mathcal{N}(Az,K(\bar{\mathbf{x}},\bar{\mathbf{x}})-AK_{y}A^{\top}), \tag{2.60}$$

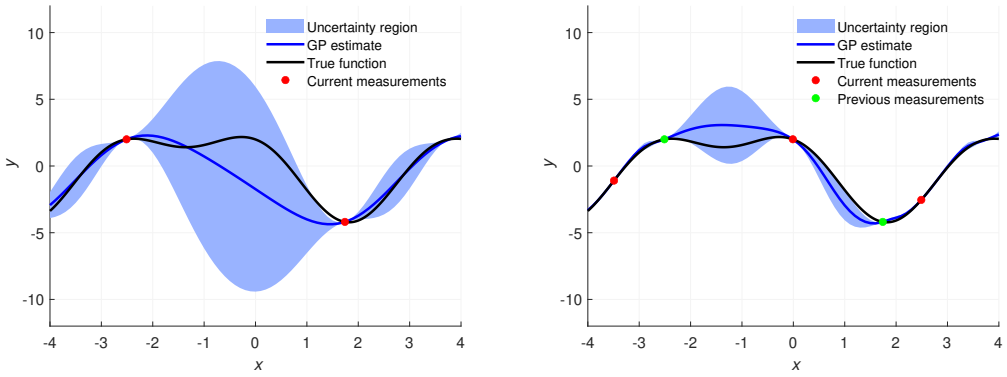
where

$$A=K(\bar{\mathbf{x}},\mathbf{x})K_{y}^{-1}, \tag{2.61}$$

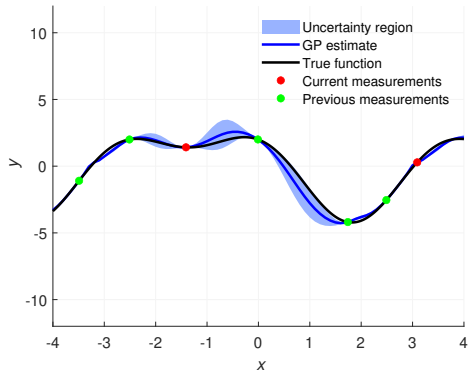
$$K_{z}=K(\mathbf{x},\mathbf{x})+I_{m}\otimes R. \tag{2.62}$$

An example to the performance of the Gaussian process is demonstrated in Figure 2.2 by a visual comparison with the linear interpolation, fitting a polynomial to estimate a function. The kernel of the Gaussian process is the periodic function 2.59 due to the periodic characteristic of the true function with the hyperparameters $\sigma_f = 3.5$ and $l = 1.5$. The Gaussian process outperforms the other techniques due to the proper kernel selection, which encodes assumptions about the function.

Although the batch approach provides both the estimate and the uncertainty, all the measurements should be available. On the other hand, extended target tracking is an excellent example of an application that collects measurements sequentially in time [4]. For such applications, recursive Gaussian regression [13, 14] makes the whole data requirement invalid, reducing the computational complexity related to the number of measurements.



(a) (b)



(c)

Figure 2.3: (a), (b) and (c) display the estimates of the function values with uncertainty region using the recursive Gaussian regression, the true function, and the measurements in successive time instants.

In the light of the assumption that $\mathbf{f}$ carries the information comes from past measurements, which makes the recursion of the state available through time k, Gaussian process assumes that z_k and $\mathbf{f}$ are jointly Gaussian such that

$$\begin{bmatrix} z_k \\ \mathbf{f} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} k\left(x_k, x_k\right) + R & K(x_k, \bar{\mathbf{x}}) \\ K(\bar{\mathbf{x}}, x_k) & K(\bar{\mathbf{x}}, \bar{\mathbf{x}}) \end{bmatrix}\right), \tag{2.63}$$

where

$$K(\bar{\mathbf{x}}, \bar{\mathbf{x}}) = \begin{bmatrix} k\left(\bar{x}_1, \bar{x}_1\right) & \dots & k\left(\bar{x}_1, \bar{x}_n\right) \\ \vdots & & \vdots \\ k(\bar{x}_n, \bar{x}_1) & \dots & k(\bar{x}_n, \bar{x}_n) \end{bmatrix}. \tag{2.64}$$

The measurement likelihood and the initial prior can be written as

$$p(z_k|\mathbf{f}) \sim \mathcal{N}(z_k; H^{\mathbf{f}}(x_k)\mathbf{f}, R^{\mathbf{f}}(x_k)), \tag{2.65a}$$

$$p(\mathbf{f}) \sim \mathcal{N}(0, K(\bar{\mathbf{x}}, \bar{\mathbf{x}})), \tag{2.65b}$$

where

$$H^{\mathbf{f}}(x_k) = K(x_k, \bar{\mathbf{x}})[K(\bar{\mathbf{x}}, \bar{\mathbf{x}})]^{-1}, \tag{2.66a}$$

$$R^{\mathbf{f}}(x_k) = k\,(x_k, x_k) + R - K(x_k, \bar{\mathbf{x}})[K(\bar{\mathbf{x}}, \bar{\mathbf{x}})]^{-1}K(\bar{\mathbf{x}}, x_k). \tag{2.66b}$$

The measurement likelihood and the initial prior in 2.65 can be incorporated to a state-space model for recursive regression. The corresponding state-space model is constructed as follows

$$\mathbf{x}_k^{\mathbf{f}} = F^{\mathbf{f}}\mathbf{x}_{k-1}^{\mathbf{f}} + w_k, \quad w_k \sim \mathcal{N}(0,\ Q^{\mathbf{f}}), \tag{2.67a}$$

$$z_k = H^{\mathbf{f}}(x_k)\mathbf{x}_k^{\mathbf{f}} + v_k^f, \quad v_k^f \sim \mathcal{N}(0,\ R^{\mathbf{f}}(x_k)), \tag{2.67b}$$

where $\mathbf{x}_k^{\mathbf{f}} = \begin{bmatrix} f(\bar{x}_1) & \cdots & f(\bar{x}_n) \end{bmatrix}^\top$. $F^{\mathbf{f}}$ and $Q^{\mathbf{f}}$ are the state transition matrix and the process noise covariance matrix, respectively, and can be described depending on the system's requirements. As an example, the same function defined in the 2.2 is estimated using the recursive Gaussian process regression in Figure 2.3 with the same kernel function and the hyperparameters.

2.5 Variational Inference

The main purpose of the variational Bayes is to represent a conditional density of latent variables given observed ones $p(z|x)$ as an approximate density $q(z)$, and the basic idea is to treat inference as an optimization problem. The objective function of this optimization problem is provided by the field of information theory, called Kullback-Leibler (KL) divergence. The KL divergence between the probability distributions q and p is formulated as

$$KL(q||p) = -\sum q(z) \log \frac{p(z|x)}{q(z)}, \tag{2.68}$$

where KL divergence measures the similarity between the probability distributions $q(z)$ and $p(z|x)$, and has the following properties:

- $KL(q||p) \geq 0$.
- $KL(q||p) = 0 \leftrightarrow q = p$.
- $KL(q||p) \neq KL(p||q)$, which states that KL divergence is not symmetric.

The KL divergence formula can be manipulated by considering the conditional density formula $p(X, Y) = p(X|Y)p(Y)$ as follows

$$
\begin{aligned}
& KL(q(z)||p(z|x)) \\
&= -\sum_z q(z) \log \frac{p(x,z)}{q(z)p(x)}, && (2.69)\\
&= -\sum_z q(z) \left(\log \frac{p(x,z)}{q(z)} + \log \frac{1}{p(x)} \right), && (2.70)\\
&= -\sum_z q(z) \log \frac{p(x,z)}{q(z)} + \log p(x) \underbrace{\sum_z q(z)}_{1}, && (2.71)\\
&= \underbrace{-\sum_z q(z) \log \frac{p(x,z)}{q(z)}}_{\mathcal{L}} + \log p(x), && (2.72)
\end{aligned}
$$

where $\mathcal{L} = -KL(q(z)||p(x,z)$, which stands for lower bound, is always negative due to the nonnegativity of KL divergence. Furthermore, $\log p(x)$, which does not depend on q, is a constant term. The final expression is given as

$$KL(q(z)||p(z|x)) + \mathcal{L} = \text{const.} \tag{2.73}$$

Based on the relation stated in (2.73), a larger $\mathcal{L}$ reduces the $KL(q(z)||p(z|x))$, which results in a better approximate density $q(z)$ of $p(z|x)$ in terms of similarity. As a result, variational inference finds $q(z)$ that maximizes the lower bound $\mathcal{L}$ to approximate $p(z|x)$. At this point, a crucial point is the description of the variational family. One popular choice is the mean-field variational family, which assumes that latent variables or vectors are independent and can be factorized as

$$q(z) = \prod_i q_i(z_i). \tag{2.74}$$

The conventional method for this variational family is to iteratively optimize the lower bound $\mathcal{L}$ over q_i while keeping the others fixed on the latest version.

CHAPTER 3

CONFLICT RESOLUTION IN MULTI-SENSOR FUSION WITH STUDENT-T DISTRIBUTIONS

Multi-sensor fusion has been a topic of interest for many years, and several algorithms have been proposed in the literature according to the architecture of the fusion center.

The direct transmission of measurements to the fusion center for processing is the foundation of the conventional strategy, *so called* centralized architecture, for fusion. The centralized architecture is theoretically the best option if the collected measurements are accessible by the fusion center. On the other hand, the transmission of measurements, a crucial step in centralized fusion, necessitates a large amount of bandwidth and delay, which is among the disadvantages of the centralized architecture. Another architecture for fusion is decentralised architecture which is constructed with multiple fusion nodes. Each fusion node fuses its local data with the information gathered from different sources. The communication issues of the decentralised architecture depend on the fusion strategy. For instance, if high number of nodes communicate with each other, the cost problem brings us scalability problems. As a further downside, a large amount of bandwidth is also required. Although optimal performance is the main strength of the centralized architecture, the decentralised approach can provide comparable results, and further advantages related to flexibility, robustness, and scalability [15]. In decentralised architecture, any node is not superior to the others, and their communication depends on the need. Although such fusion is conceptually more complicated, centralized fusion increases the susceptibility compared to decentralised fusion in case of failure. Furthermore, decentralised architecture comes up with lower communication and processing load due to its modular structure. The flexible, modular, and robust nature of the decentralised fusion makes

it a compelling option [16, 17]. In decentralised fusion, an important source of correlation is the process noise in the dynamic state equation of the targets. Earlier works assume this noise term to be zero [16, 17]. This assumption ignores the dependency between the tracks due to common process noise. The method proposed in [18] takes this dependence into account and computes the covariance between the estimates of different sensors for two tracks. [19] proposes a formula that combines the local estimates from each sensor and considers the dependence between estimates. [20] states that the proposed formula only shows optimal performance in terms of maximum-likelihood sense. Furthermore, [20] compares the performance of the algorithms proposed by [17] and [19]. The results in [20] are constructed with a specific number of sensors. A generalized version in terms of the number of sensors is presented in [21] and [22]. [23] also derives an algorithm for an arbitrary number of sensors. Moreover, [23] shows that the performance of decentralised approach degrades more than the centralized one as the number of sensors increases. On the other hand, this performance degradation is acceptable due to the advantages provided by decentralised fusion. These theoretical developments in the decentralised fusion introduce the subject related to the applicability of decentralised fusion to practical systems. Based on this concern, [24] investigates scalable fusion algorithms. Furthermore, [24] states that the channel fusion approach [25, 26], Chernoff and Bhattacharyya fusion algorithms perform reasonably well with minimum communication.

Another categorization is based on the inputs as discussed in [27], which reviews and compares various algorithms. In their work, the authors divide the solution approaches into different categories such as measurement fusion and track fusion. In measurement fusion, the sensor measurements are fused to obtain a combined measurement. [28] compares the two commonly used measurement fusion methods for Kalman filter based multi-sensor data fusion. These methods are also studied in [29, 30] and [31]. Complex combination (CC), Bar-Shalom/Campo state vector combination (BC) [19], the best linear unbiased estimation (BLUE) [32, 33, 34] and covariance intersection (CI) are some well known methods for track fusion. Covariance intersection technique proposed by [1, 2, 3] provides more robust combination of local estimates than the linear combination. The weight calculation is an essential criterion on the performance of the CI approach. [2] and [3] calculate weights by min-

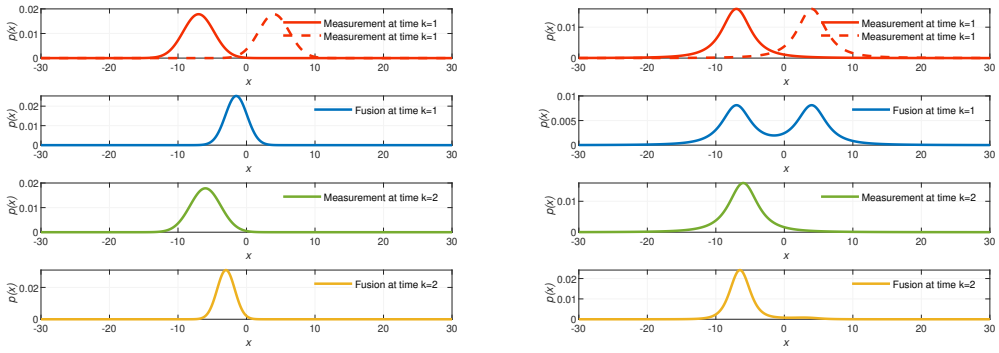
imizing the determinant of the fused mean-square error matrix, which corresponds to a costly nonlinear optimization problem. In order to decrease the computational complexity, fast CI algorithms [35, 36] calculate the weights approximately rather than the optimization of the nonlinear cost function. [37] considers a set-theoretic interpretation of CI algorithm and obtains a tighter bound on the error covariance. [10] [11] propose the generalization of CI technique beyond Gaussian distribution. This generalization leads to the Chernoff fusion of Bernoulli, Poisson and independent identically decentralised processes in [38]. Similarly, an extension of CI to Gaussian mixture models (GMM) is provided in [39]. [39] applies CI to each pair of the Gaussian components in the two mixtures. The method suggested in [39] is followed by another GMM-based approach in [40]. [40] approximates the power of a Gaussian mixture as a combination of the power of its constituents. In [41], an arbitrary power of a Gaussian mixture is approximated as an unnormalized Gaussian mixture, whose weights are the only unknown variables. These weights are calculated by using the sigma-point approximation of the Gaussian mixture to solve a weighted least squares problem.

Another essential aspect of multi-sensor fusion is robustness in case of outliers, which may originate from unreliable sensors. The investigation of alternative noise models is essential since the Gaussian assumption is insufficient to handle these noise characteristics. The heavy-tailed structure makes student-t distribution an excellent option to represent noise in filtering and fusion applications. [42] suggests a nonlinear filter that achieves more reliable performance in the presence of outliers and model mismatches by modeling both process and measurement noise as student-t distributions. [43] employs variational Bayes to create a robust and analytically recursive technique while representing non-Gaussian measurement noise as a student-t distribution for data fusion. [44] models the process and measurement noise as student-t distribution and describes an indicator variable based on Bernoulli prior for outlier rejection. A student-t hierarchical Gaussian state-space model combined with variational Bayes is constructed for parameter identification and state estimation in a centralized architecture. [45] describes a fusion algorithm that estimates state utilizing measurements sequentially and incorporates the multivariate t-decentralised process and measurement noises.

This section focuses on the idea of combining the Chernoff fusion with student t-distribution. Simulations and real data experiments are conducted to evaluate the performance.

3.1 Motivation

The motivation behind the use of student-t distributions for multi-sensor fusion can be illustrated with a toy example in Figure 3.1. In this example, two sensors provide conflicting information about the unknown variable x, represented by the probability density functions $p_{x,1}(x)$ and $p_{x,2}(x)$ at time $t = 1$, respectively. Another measurement consistent with the first sensor data is received at time $t = 2$ to resolve the uncertainty. Two different cases are compared where the underlying distributions are assumed to be Gaussian and student-t, respectively. In the Gaussian assumption, the uni-modal structure of the fused density results in an information loss at the end of the fusion since the conflicting information collapses without solving ambiguity. On the other hand, the student-t assumption compensates for the second sensor data error at $t = 1$ thanks to its multi-modal structure. The fused probability density functions with Gaussian and student-t cases are given in Figure 3.1(a) and Figure 3.1(a), respectively. In the Gaussian case, the procedure to fuse Gaussian posteriors turns out to be weighting the means and covariances of the posteriors. Hence, the fused density at $t = 1$ can be calculated according to (2.54). This weighting operation ignores the essential points of the sensor output posteriors. When the new information is processed through the fused density at $t = 1$, the overall result is misleading. Conversely, the approximation of $p_{x,1}(x)$ and $p_{x,2}(x)$ as a mixture of student-t distributions at the end of fusion captures critical information regarding the posteriors, which provides excellent flexibility to eliminate ambiguity using new data.



(a) (b)

Figure 3.1: Fusion inputs and outputs in a probabilistic manner. (a) displays the fusion inputs and outputs under the Gaussian input assumption. (b) depicts the whole procedure's inputs and outputs, with the inputs represented by student-t distribution.

3.2 Chernoff Fusion with Student-t Distributions

Chernoff fusion enables us to identify and remove common information while fusing data from two different sensors. Gaussian input probability density functions yield the same form as the Chernoff fusion inputs, which is not the case for student t-distribution use. On the other hand, the robustness in the case of student t-distribution makes adapting student t-distribution to the Chernoff fusion a compelling option. Performing Chernoff fusion with student t-distribution as inputs involves four main steps:

- taking a non-integer power of a student-t distribution;
- taking a non-integer power of a student-t mixture;
- multiplying two student-t distributions;
- merging student-t mixtures.

In this section, we investigate each of these tasks in the following subsections and propose approximations involving sigma-points when necessary.

3.2.1 A Non-Integer Power of a Student-t Distribution

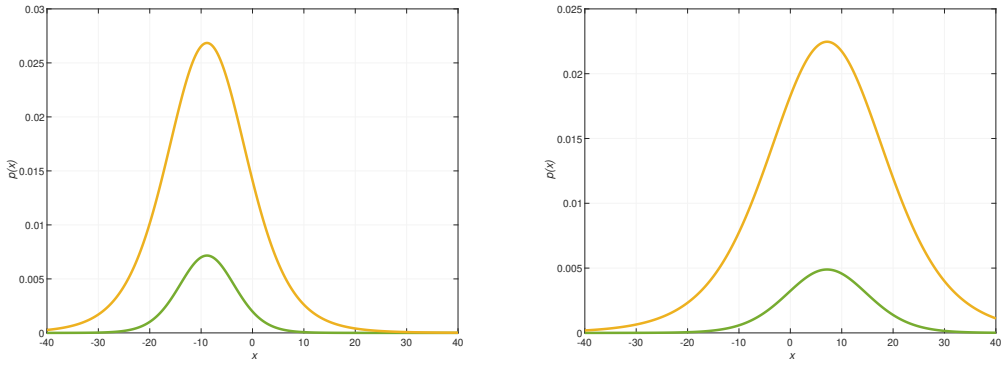
We start with an arbitrary non-negative power of a student-t distribution, which constructs the basis of our assumptions in the following subsections. An arbitrary non-negative power of a student-t distribution can be written as an unnormalized student-t distribution as follows

$$
\begin{aligned}
& t_{\nu}^{w}(x;\mu,\Sigma) \\
=& \begin{pmatrix} \frac{\Gamma^w((\nu+n)/2)}{\Gamma^w(\nu/2)|\pi\nu\Sigma|^{w/2}} \\ \times \left(1+(x-\mu)^{\mathrm{T}}(\nu\Sigma)^{-1}(x-\mu)\right)^{-w(\nu+n)/2} \end{pmatrix} && (3.1) \\
=& \begin{pmatrix} \frac{\Gamma^w((\nu+n)/2)}{\Gamma^w(\nu/2)|\pi\nu\Sigma|^{(w-1)/2}} \frac{1}{|\pi\nu\Sigma|^{1/2}} \\ \times \left(1+(x-\mu)^{\mathrm{T}}(\nu\Sigma)^{-1}(x-\mu)\right)^{-(w\nu+(w-1)n+n)/2} \end{pmatrix} && (3.2) \\
=& \begin{pmatrix} \frac{\Gamma^w((\nu+n)/2)}{\Gamma^w(\nu/2)|\pi\nu\Sigma|^{(w-1)/2}} \frac{1}{\left|\pi\bar{\nu}\frac{\nu}{\bar{\nu}}\Sigma\right|^{1/2}} \\ \times \left(1+(x-\mu)^{\mathrm{T}}\left(\bar{\nu}\frac{\nu}{\bar{\nu}}\Sigma\right)^{-1}(x-\mu)\right)^{-(\bar{\nu}+n)/2} \end{pmatrix} && (3.3) \\
=& \begin{pmatrix} \frac{\Gamma^w((\nu+n)/2)}{\Gamma^w(\nu/2)|\pi\nu\Sigma|^{(w-1)/2}} \frac{1}{\left|\pi\bar{\nu}\overline{\Sigma}\right|^{1/2}} \\ \times \left(1+(x-\mu)^{\mathrm{T}}\left(\bar{\nu}\overline{\Sigma}\right)^{-1}(x-\mu)\right)^{-(\bar{\nu}+n)/2} \end{pmatrix} && (3.4) \\
=& \begin{pmatrix} \frac{\Gamma^w((\nu+n)/2)\Gamma(\bar{\nu}/2)}{\Gamma^w(\nu/2)\Gamma((\bar{\nu}+n)/2)|\pi\nu\Sigma|^{(w-1)/2}} \frac{\Gamma((\bar{\nu}+n)/2)}{\Gamma(\bar{\nu}/2)\left|\pi\bar{\nu}\overline{\Sigma}\right|^{1/2}} \\ \times \left(1+(x-\mu)^{\mathrm{T}}\left(\bar{\nu}\overline{\Sigma}\right)^{-1}(x-\mu)\right)^{-(\bar{\nu}+n)/2} \end{pmatrix} \\
=& c_{\nu}(w,\Sigma)t_{\bar{\nu}}(x;\mu,\overline{\Sigma}), && (3.5)
\end{aligned}
$$

where

$$
\begin{aligned}
\bar{\nu} \triangleq & w(\nu+n)-n, && (3.6a) \\
\overline{\Sigma} \triangleq & \frac{\nu}{\bar{\nu}}\Sigma, && (3.6b) \\
c_{\nu}(w,\Sigma) \triangleq & \frac{\Gamma^w((\nu+n)/2)\Gamma(\bar{\nu}/2)}{\Gamma^w(\nu/2)\Gamma((\bar{\nu}+n)/2)|\pi\nu\Sigma|^{(w-1)/2}}. && (3.6c)
\end{aligned}
$$

Two realizations for the power of a student-t distribution with the exponent factor 0.5 are provided in Figure 3.2.



(a) (b)

Figure 3.2: (a) and (b) display two realizations for the power of a student-t distribution with the exponent factor of 0.5. The power of the green student t-distributions with the exponent factor 0.5 outputs the corresponding yellow distributions.

3.2.2 A Non-Integer Power of a Student-t Mixture

A student-t mixture model can be written as the sum of the weighted student-t distributions as follows

$$p(x) = \sum_{i=1}^{N} \alpha_i t_{\nu_i}(x; \mu_i, \Sigma_i). \tag{3.7}$$

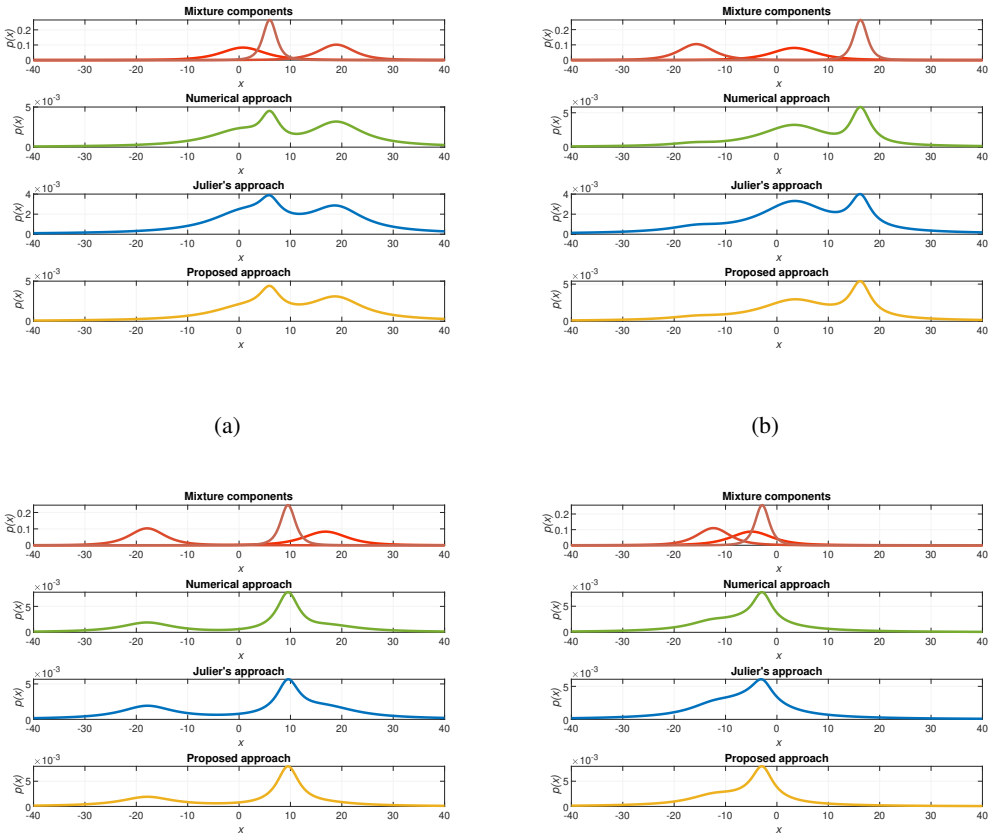
The wth power of the student-t mixture is essential, as shown in the Chernoff fusion formula (2.52). Although this procedure is straightforward for the student-t distribution, an approximation is required for the student-t mixture case since a non-integer power of a student-t mixture is not a student-t mixture in general. Hence we propose the following approximation based on the non-integer power expressions for a single student-t distribution given in the previous section

$$p^w(x) \approx q_\beta(x) \triangleq \sum_{i=1}^{N} \beta_i t_{\bar{\nu}_i}(x; \mu_i, \overline{\Sigma}_i) \tag{3.8}$$

where $\beta_i \geq 0$, $1 \leq i \leq N$, are some unknown weights and

$$\bar{\nu}_i \triangleq w(\nu_i + n) - n, \tag{3.9a}$$

$$\overline{\Sigma}_i \triangleq \frac{\nu_i}{\bar{\nu}_i} \Sigma_i. \tag{3.9b}$$



(a) (b)

(c) (d)

Figure 3.3: (a-d) display four realizations of the power of a student-t mixture with the exponent factor of 0.5 for different methods. Red distributions are the individual components of the mixture. Green, blue and yellow distributions illustrate the power of a student-t mixture determined by numerical, Julier's, and proposed approaches, respectively.

We propose to find the unknown weights $\beta = [\beta_1, \beta_2, \ldots, \beta_N]$ by solving the following optimization problem as in [41],

$$\hat{\beta} = \arg\min_{\beta \geq 0} \int (p^w(x) - q_\beta(x))^2 p(x)\, \mathrm{d}x. \tag{3.10}$$

Since an analytical approach to determine the $p^w(x)$ is not feasible, it is not possible to evaluate the cost function above without resorting to numerical methods, which would be too computationally costly. We here use the concept of sigma-points for this problem as in [41]. Suppose that we have the following sigma-point approximation

for the ith mixture component $t_{\nu_i}(x;\mu_i,\Sigma_i)$ of $q_\beta(x)$

$$t_{\nu_i}(x;\mu_i,\Sigma_i)=\sum_{j=1}^{L}\pi_i^{(j)}\delta_{s_i^{(j)}}(x), \tag{3.11}$$

where $\{s_i^{(j)}\}_{j=1}^{L}$, $\{\pi_i^{(j)}\}_{j=1}^{L}$ are the sigma-points and their weights. Restatement of the $p(\cdot)$ using the sigma-point approximation in (3.11) gives the following overall approximation

$$p(x)=\sum_{i=1}^{N}\alpha_i\sum_{j=1}^{L}\pi_i^{(j)}\delta_{s_i^{(j)}}(x). \tag{3.12}$$

After substituting this approximation, the cost function in (3.10) can be modified as

$$\begin{aligned}&\int (p^w(x)-q_\beta(x))^2p(x)\,\mathrm{d}x,\\ &\quad\approx\sum_{i=1}^{N}\sum_{j=1}^{L}\alpha_i\pi_i^{(j)}\Big(p^w\big(s_i^{(j)}\big)-q_\beta\big(s_i^{(j)}\big)\Big)^2,\end{aligned} \tag{3.13}$$

which allows us to rewrite the cost function in a more compact way

$$\int (p^w(x)-q_\beta(x))^2p(x)\,\mathrm{d}x\approx(A\beta-b)^\top W(A\beta-b), \tag{3.14}$$

where the elements of the vector $b\in\mathbb{R}^{NL\times 1}$, the matrix $A\in\mathbb{R}^{NL\times N}$ and the diagonal matrix $W\in\mathbb{R}^{NL\times NL}$ are defined as

$$[A]_{L(i-1)+j,m}\triangleq t_{\bar{\nu}_m}\left(s_i^{(j)};\mu_m,\overline{\Sigma}_m\right), \tag{3.15}$$

$$[b]_{L(i-1)+j,1}\triangleq p^w(s_i^{(j)}), \tag{3.16}$$

$$[W]_{L(i-1)+j,L(i-1)+j}=\alpha_i\pi_i^{(j)}, \tag{3.17}$$

for $i,m=1,\ldots,N$ and $j=1,\ldots,L$. As a consequence, the optimization in (3.10) turns into the following

$$\hat{\beta}\triangleq\arg\min_{\beta\geq 0}(A\beta-b)^\top W(A\beta-b), \tag{3.18}$$

which is a non-negative weighted least squares problem. The so-called Lawson-Hanson algorithm can solve this problem, and its modified version is implemented in the built-in Matlab function as explained in [41]. In order to apply Lawson-Hanson algorithm, the problem in (3.18) is converted to the equivalent version as follows

$$\hat{\beta}\triangleq\arg\min_{\beta\geq 0}\|\tilde{A}\beta-\tilde{b}\|_2^2, \tag{3.19a}$$

$$\tilde{A}\triangleq W^{1/2}A,\qquad \tilde{b}\triangleq W^{1/2}b. \tag{3.19b}$$

The approach in [40] aims to extend CI to Gaussian mixture models. For this purpose, the non-integer power of Gaussian mixture is expressed using the following approximation

$$\left(\sum_{i=1}^{n} p_i(x)\right)^{\omega} \approx \sum_{i=1}^{n} p_i^{\omega}(x). \tag{3.20}$$

Figure 3.3 provides different realizations of the numerical approach, the method in [40], and the proposed approach in addition to the individual components of the mixture. The approximation in (3.20) is used to determine the non-integer power of the student t-mixture and labeled as Julier's approach in Figure 3.3. According to Figure 3.3, the proposed approach is a better approximation than Julier's approach in terms of the similarity with the numerical approach.

3.2.3 Multiplication of Two Student-t Distributions

Suppose that we have two student-t distributions, namely, $p_1(x) \triangleq t_{\nu_1}(x, \mu_1, \Sigma_1)$ and $p_2(x) \triangleq t_{\nu_2}(x, \mu_2, \Sigma_2)$. Since the multiplication $p_1(x)p_2(x)$ is not a student-t distribution in general, we would like to approximate the multiplication as a student-t mixture as follows

$$p_1(x)p_2(x) \approx q_\eta(x) \triangleq \eta_1 p_1(x) + \eta_2 p_2(x). \tag{3.21}$$

Note that the only unknowns in this approximation are the weights $\eta \triangleq [\eta_1\ ,\ \eta_2]$. We propose to find these unknown weights by using the following optimization problem,

$$\hat{\eta} \triangleq \arg\min_{\eta \geq 0} \int (p_1(x)p_2(x) - q_\eta(x))^2 (p_1(x) + p_2(x))\ \mathrm{d}x. \tag{3.22}$$

Since $p_1(x)p_2(x)$ is not available analytically, we consider numerical approach as in the previous section. On the other hand, evaluating any possible η, which increases computational load too much, leads to the following sigma-point approximations for the densities $p_1(\cdot)$ and $p_2(\cdot)$

$$p_1(x) \approx \sum_{i=1}^{L} \pi_1^{(i)} \delta_{s_1^{(i)}}(x), \tag{3.23a}$$

$$p_2(x) \approx \sum_{i=1}^{L} \pi_2^{(i)} \delta_{s_2^{(i)}}(x), \tag{3.23b}$$

where $\{s_1^{(i)}\}_{i=1}^L$, $\{\pi_1^{(i)}\}_{i=1}^L$ and $\{s_2^{(i)}\}_{i=1}^L$, $\{\pi_2^{(i)}\}_{i=1}^L$ are the sigma-points and their weights representing $p_1(\cdot)$ and $p_2(\cdot)$, respectively. Substituting the approximations into to the cost function (3.22) gives the following

$$\begin{aligned}
&\int (p_1(x)p_2(x) - q_\eta(x))^2(p_1(x) + p_2(x))\,\mathrm{d}x \\
&= \int (p_1(x)p_2(x) - q_\eta(x))^2 p_1(x)\,\mathrm{d}x + \int (p_1(x)p_2(x) - q_\eta(x))^2 p_2(x)\,\mathrm{d}x, \quad (3.24a)\\
&\approx \sum_{i=1}^{L} \pi_1^{(i)} \Big(p_1\big(s_1^{(i)}\big)p_2\big(s_1^{(i)}\big) - q_\eta\big(s_1^{(i)}\big)\Big)^2 + \sum_{i=1}^{L} \pi_2^{(i)} \Big(p_1\big(s_2^{(i)}\big)p_2\big(s_2^{(i)}\big) - q_\eta\big(s_2^{(i)}\big)\Big)^2, \quad (3.24b)\\
&= (A_1\eta - b_1)^\top W_1 (A_1\eta - b_1) + (A_2\eta - b_2)^\top W_2 (A_2\eta - b_2), \quad (3.24c)
\end{aligned}$$

where $A_1, A_2 \in \mathbb{R}^{L\times 2}$, $b_1, b_2 \in \mathbb{R}^L$ and $W_1, W_2 \in \mathbb{R}^{L\times L}$ are defined as follows

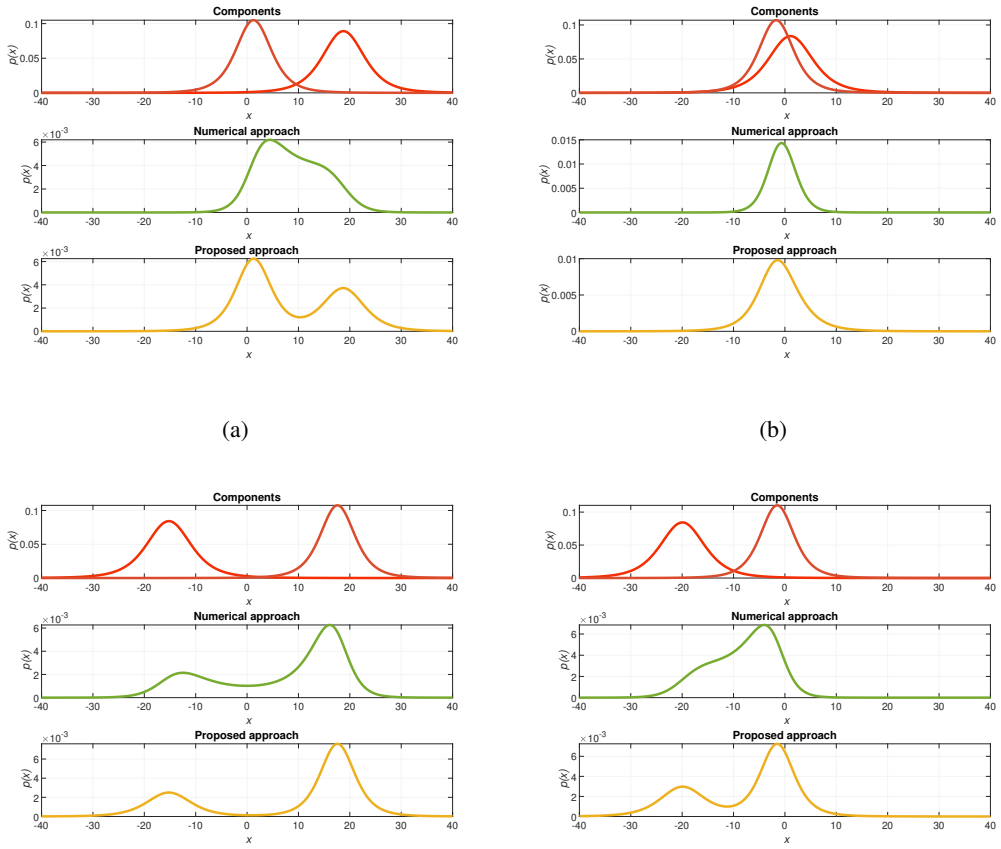
$$\begin{aligned}
[A_1]_{i,j} &\triangleq p_j\big(s_1^{(i)}\big), && (3.25a)\\
[A_2]_{i,j} &\triangleq p_j\big(s_2^{(i)}\big), && (3.25b)\\
[b_1]_i &\triangleq p_1\big(s_1^{(i)}\big)p_2\big(s_1^{(i)}\big), && (3.25c)\\
[b_2]_i &\triangleq p_1\big(s_2^{(i)}\big)p_2\big(s_2^{(i)}\big), && (3.25d)\\
[W_1]_{i,m} &\triangleq \begin{cases} \pi_1^{(i)}, & i = m \\ 0, & \text{otherwise} \end{cases}, && (3.25e)\\
[W_2]_{i,m} &\triangleq \begin{cases} \pi_2^{(i)}, & i = m \\ 0, & \text{otherwise} \end{cases}, && (3.25f)
\end{aligned}$$

for $i, m = 1, \ldots, N$, $j = 1, \ldots, L$. Introducing sigma-point approximations lets us write the cost function in a vector form

$$\int (p_1(x)p_2(x) - q_\eta(x))^2(p_1(x) + p_2(x))\,\mathrm{d}x \approx (A\eta - b)^\top W (A\eta - b), \quad (3.26)$$

where the augmented quantities $A \in \mathbb{R}^{2L\times 2}$, $b \in \mathbb{R}^{2L}$ and $W \in \mathbb{R}^{2L\times 2L}$ are defined as

$$\begin{aligned}
A &\triangleq \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}, && (3.27a)\\
b &\triangleq \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, && (3.27b)\\
W &\triangleq \mathrm{blkdiag}(W_1, W_2). && (3.27c)
\end{aligned}$$



(a) (b)

(c) (d)

Figure 3.4: Four different realizations of the multiplication of two student-t distributions. Red distributions are the individual components of the mixture. The outcomes of multiplication calculated by numerical and proposed procedures are represented by green and yellow distributions, respectively.

Hence, the optimization in (3.22) can be written as follows

$$\hat{\eta} \triangleq \arg\min_{\eta \geq 0} (A\eta - b)^\top W (A\eta - b), \tag{3.28}$$

which is again a non-negative weighted least squares problem which can be solved as explained in the previous section. Various realizations of the numerical and proposed approaches are provided in Figure 3.4, along with the individual components.

3.2.4 Merging Student-t Mixtures

This section presents a merging strategy to maintain the final form of the Chernoff fusion procedure as a student-t distribution. Otherwise, a more significant number of components in the student-t mixture is expected over time, which increases computational cost. Based on the merging of the components, we would like to approximate a student-t mixture as in (3.7) with a single student-t distribution defined in (2.42). Kullback-Leibler (KL) is the preferred method to find the approximation

$$\hat{q}(\cdot) = \arg \min_{q(x) \triangleq t_\nu(x;\mu,\Sigma)} \mathrm{KL}(p(\cdot)||q(\cdot)), \tag{3.29a}$$

$$= \arg \max_{q(x) \triangleq t_\nu(x;\mu,\Sigma)} E_p\left[\log q(x)\right], \tag{3.29b}$$

where $E_p[\cdot]$ denotes the expectation operation with respect to the mixture $p(\cdot)$ and $\log q(\cdot)$ is given as

$$\begin{aligned} &\log t_\nu(x;\mu,\Sigma) \\ &\stackrel{+}{=} \log\Gamma\left(\frac{\nu+n}{2}\right) - \log\Gamma\left(\frac{\nu}{2}\right) - \frac{n}{2}\log\nu - \frac{1}{2}\log|\Sigma| \\ &\quad - \frac{\nu+n}{2}\log\left(1+(x-\mu)^{\mathrm{T}}(\nu\Sigma)^{-1}(x-\mu)\right), \end{aligned} \tag{3.30}$$

where the sign $\stackrel{+}{=}$ denotes equality up to an additive constant. The solution of the optimization problem (3.29b) is based on a joint approach, which starts with equating the derivatives with respect to the optimized variables ν, μ and Σ to zero.

- **Maximizing $E_p\left[\log q(x)\right]$ with respect to ν:**

$$2\frac{\partial}{\partial\nu}E_p\left[\log q(x)\right] = 0, \tag{3.31a}$$

$$\begin{aligned} &\psi\left(\frac{\nu+n}{2}\right) - E_p\left[\log\left(1+(x-\mu)^{\mathrm{T}}(\nu\Sigma)^{-1}(x-\mu)\right)\right] \\ &+\left(\frac{\nu+n}{\nu}\right)E_p\left[\frac{(x-\mu)^{\mathrm{T}}(\nu\Sigma)^{-1}(x-\mu)}{1+(x-\mu)^{\mathrm{T}}(\nu\Sigma)^{-1}(x-\mu)}\right] - \psi\left(\frac{\nu}{2}\right) - \frac{n}{\nu} = 0, \end{aligned} \tag{3.31b}$$

  where ψ denotes the logarithmic Gamma function defined as $\psi(x) \triangleq \frac{\mathrm{d}}{\mathrm{d}x}\log\Gamma(x)$. By adjusting the equation (3.31b), we have the following

$$\nu = n\frac{1 - E_p\left[\frac{(x-\mu)^{\mathrm{T}}(\nu\Sigma)^{-1}(x-\mu)}{1+(x-\mu)^{\mathrm{T}}(\nu\Sigma)^{-1}(x-\mu)}\right]}{\begin{pmatrix}\psi\left(\frac{\nu+n}{2}\right) - \psi\left(\frac{\nu}{2}\right) + E_p\left[\frac{(x-\mu)^{\mathrm{T}}(\nu\Sigma)^{-1}(x-\mu)}{1+(x-\mu)^{\mathrm{T}}(\nu\Sigma)^{-1}(x-\mu)}\right] \\ -E_p\left[\log\left(1+(x-\mu)^{\mathrm{T}}(\nu\Sigma)^{-1}(x-\mu)\right)\right]\end{pmatrix}}. \tag{3.32}$$

- **Maximizing $E_p\left[\log q(x)\right]$ with respect to μ:**

$$\frac{\partial}{\partial \mu} E_p\left[\log q(x)\right] = 0, \tag{3.33a}$$

$$2(\nu+n) E_p\left[\frac{(\nu\Sigma)^{-1}(x-\mu)}{1+(x-\mu)^{\mathrm{T}}(\nu\Sigma)^{-1}(x-\mu)}\right] = 0\ , \tag{3.33b}$$

which can be written as follows

$$\mu = \frac{E_p\left[\frac{x}{1+(x-\mu)^{\mathrm{T}}(\nu\Sigma)^{-1}(x-\mu)}\right]}{E_p\left[\frac{1}{1+(x-\mu)^{\mathrm{T}}(\nu\Sigma)^{-1}(x-\mu)}\right]}. \tag{3.34}$$

- **Maximizing $E_p\left[\log q(x)\right]$ with respect to Σ:** Taking derivative with respect to Σ or Σ^{-1} is equivalent for optimization purposes since Σ is assumed to be invertible. For this purpose, we use the following matrix calculus results

$$\frac{\mathrm{d}\log|X|}{\mathrm{d}X} = X^{-1}, \tag{3.35a}$$

$$\frac{\mathrm{d}x^{\mathrm{T}}Xx}{\mathrm{d}X} = xx^{\mathrm{T}}, \tag{3.35b}$$

for a symmetric matrix X. Based on the (3.35), we have the following result

$$2\frac{\partial}{\partial \Sigma^{-1}} E_p\left[\log q(x)\right] = 0, \tag{3.36a}$$

$$\Sigma - \left(\frac{\nu+n}{\nu}\right) E_p\left[\frac{(x-\mu)(x-\mu)^{\mathrm{T}}}{1+(x-\mu)^{\mathrm{T}}(\nu\Sigma)^{-1}(x-\mu)}\right] = 0, \tag{3.36b}$$

which can be rearranged as follows

$$\Sigma = \frac{\nu+n}{\nu} E_p\left[\frac{(x-\mu)(x-\mu)^{\mathrm{T}}}{1+(x-\mu)^{\mathrm{T}}(\nu\Sigma)^{-1}(x-\mu)}\right]. \tag{3.37}$$

An analytical solution to these equations is not available, whose drawback is solving them using the following fixed-point iteration technique

$$\nu^{[j+1]} = \frac{n\left(1 - E_p\left[\frac{(x-\mu^{[j]})^{\mathrm{T}}(\nu^{[j]}\Sigma^{[j]})^{-1}(x-\mu^{[j]})}{1+(x-\mu^{[j]})^{\mathrm{T}}(\nu^{[j]}\Sigma^{[j]})^{-1}(x-\mu^{[j]})}\right]\right)}{\left(\begin{array}{c}\psi\left(\frac{\nu^{[j]}+n}{2}\right) - \psi\left(\frac{\nu^{[j]}}{2}\right) \\ -E_p\left[\log\left(1+(x-\mu^{[j]})^{\mathrm{T}}(\nu^{[j]}\Sigma^{[j]})^{-1}(x-\mu^{[j]})\right)\right] \\ +E_p\left[\frac{(x-\mu^{[j]})^{\mathrm{T}}(\nu^{[j]}\Sigma^{[j]})^{-1}(x-\mu^{[j]})}{1+(x-\mu^{[j]})^{\mathrm{T}}(\nu^{[j]}\Sigma^{[j]})^{-1}(x-\mu^{[j]})}\right]\end{array}\right)}, \tag{3.38a}$$

$$\mu^{[j+1]} = \frac{E_p\left[\frac{x}{1+(x-\mu^{[j]})^{\mathrm{T}}(\nu^{[j]}\Sigma^{[j]})^{-1}(x-\mu^{[j]})}\right]}{E_p\left[\frac{1}{1+(x-\mu^{[j]})^{\mathrm{T}}(\nu^{[j]}\Sigma^{[j]})^{-1}(x-\mu^{[j]})}\right]}, \tag{3.38b}$$

$$\Sigma^{[j+1]} = \frac{\nu^{[j]}+n}{\nu^{[j]}} E_p\left[\frac{(x-\mu^{[j]})(x-\mu^{[j]})^{\mathrm{T}}}{1+(x-\mu^{[j]})^{\mathrm{T}}(\nu^{[j]}\Sigma^{[j]})^{-1}(x-\mu^{[j]})}\right], \tag{3.38c}$$

where the superscripts with brackets denote the iteration number and the initial values of the iteration are selected as follows

$$\nu^{[0]} = \sum_{i=1}^{N} \alpha_i \nu_i, \tag{3.39a}$$

$$\mu^{[0]} = \sum_{i=1}^{N} \alpha_i \mu_i, \tag{3.39b}$$

$$\Sigma^{[0]} = \sum_{i=1}^{N} \alpha_i \left(\frac{\nu_i}{\nu_i - 2}\Sigma_i + (\mu_i - \mu^{[0]})(\mu_i - \mu^{[0]})^{\mathrm{T}}\right). \tag{3.39c}$$

The expectations appearing in the iterations above are proposed to be taken using the sigma-points generated for each component of $p(\cdot)$ as given in [46, 47]. We generate sigma points for each component of $p(\cdot)$ as

$$p(x) \approx \sum_{i=1}^{N} \alpha_i \underbrace{\sum_{\ell=1}^{L} \pi_i^{(\ell)} \delta_{x_i^{(\ell)}}(x)}_{\approx t_{\nu_i}(x;\mu_i,\Sigma_i)}, \tag{3.40}$$

where $\{x_i^{(\ell)}\}_{\ell=1}^{L}$ and $\{\pi_i^{(\ell)}\}_{\ell=1}^{L}$ denote the sigma-points and their weights for the ith mixture component. Then, the expectations appearing in the iterations above can be

calculated as follows

$$E_p\Big[\log\Big(1+(x-\mu^{[j]})^{\mathrm{T}}(\nu^{[j]}\Sigma^{[j]})^{-1}(x-\mu^{[j]})\Big)\Big]$$
$$=\sum_{i=1}^{N}\alpha_i\sum_{\ell=1}^{L}\pi_i^{(\ell)}\log\left(\mathcal{T}\left(x_i^{(\ell)},\mu^{[j]},\nu^{[j]},\Sigma^{[j]}\right)\right), \tag{3.41a}$$

$$E_p\left[\frac{(x-\mu^{[j]})^{\mathrm{T}}(\nu^{[j]}\Sigma^{[j]})^{-1}(x-\mu^{[j]})}{1+(x-\mu^{[j]})^{\mathrm{T}}(\nu^{[j]}\Sigma^{[j]})^{-1}(x-\mu^{[j]})}\right]$$
$$=\sum_{i=1}^{N}\alpha_i\sum_{\ell=1}^{L}\frac{\pi_i^{(\ell)}(x_i^{(\ell)}-\mu^{[j]})^{\mathrm{T}}(\nu^{[j]}\Sigma^{[j]})^{-1}(x_i^{(\ell)}-\mu^{[j]})}{1+(x_i^{(\ell)}-\mu^{[j]})^{\mathrm{T}}(\nu^{[j]}\Sigma^{[j]})^{-1}(x_i^{(\ell)}-\mu^{[j]})}, \tag{3.41b}$$

$$E_p\left[\frac{x}{1+(x-\mu^{[j]})^{\mathrm{T}}(\nu^{[j]}\Sigma^{[j]})^{-1}(x-\mu^{[j]})}\right]$$
$$=\sum_{i=1}^{N}\alpha_i\sum_{\ell=1}^{L}\frac{\pi_i^{(\ell)}x_i^{(\ell)}}{1+(x_i^{(\ell)}-\mu^{[j]})^{\mathrm{T}}(\nu^{[j]}\Sigma^{[j]})^{-1}(x_i^{(\ell)}-\mu^{[j]})}, \tag{3.41c}$$

$$E_p\left[\frac{1}{1+(x-\mu^{[j]})^{\mathrm{T}}(\nu^{[j]}\Sigma^{[j]})^{-1}(x-\mu^{[j]})}\right]$$
$$=\sum_{i=1}^{N}\alpha_i\sum_{\ell=1}^{L}\frac{\pi_i^{(\ell)}}{1+(x_i^{(\ell)}-\mu^{[j]})^{\mathrm{T}}(\nu^{[j]}\Sigma^{[j]})^{-1}(x_i^{(\ell)}-\mu^{[j]})}, \tag{3.41d}$$

$$E_p\left[\frac{(x-\mu^{[j]})(x-\mu^{[j]})^{\mathrm{T}}}{1+(x-\mu^{[j]})^{\mathrm{T}}(\nu^{[j]}\Sigma^{[j]})^{-1}(x-\mu^{[j]})}\right]$$
$$=\sum_{i=1}^{N}\alpha_i\sum_{\ell=1}^{L}\frac{\pi_i^{(\ell)}(x_i^{(\ell)}-\mu^{[j]})(x_i^{(\ell)}-\mu^{[j]})^{\mathrm{T}}}{1+(x_i^{(\ell)}-\mu^{[j]})^{\mathrm{T}}(\nu^{[j]}\Sigma^{[j]})^{-1}(x_i^{(\ell)}-\mu^{[j]})}, \tag{3.41e}$$

where the operator $\mathcal{T}(\cdot)$ given in (3.41a) is defined as

$$\mathcal{T}(A,B,C,D)=1+(A-B)^{\mathrm{T}}(CD)^{-1}(A-B). \tag{3.42}$$

Various realizations of the proposed approach are provided in Figure 3.5, along with the individual components.

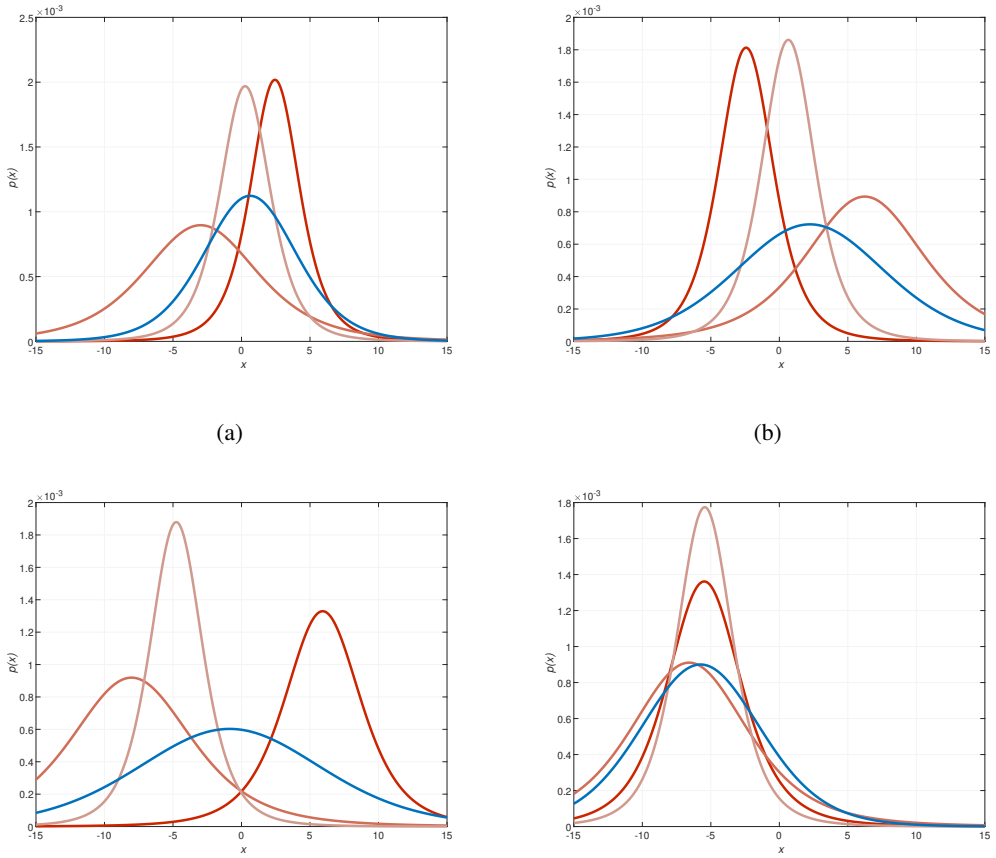


Figure 3.5: (a-d) display four realizations of the merging student-t mixture. Distributions with different shades of red are the individual components of the mixture. The blue distribution illustrates the outcome of the merging procedure. From the individual component with the darkest shade of red to the one with the lightest shade, the weights are as given: (a) $\rightarrow \alpha$ = [0.2975, 0.1761, 0.5264], (b) $\rightarrow \alpha$ = [0.2014, 0.4682, 0.3304], (c) $\rightarrow \alpha$ = [0.4227, 0.1926, 0.3847], (d) $\rightarrow \alpha$ = [0.5222, 0.3077, 0.1701].

3.2.5 Determination of Exponent Factors

For the Chernoff fusion-based methods (CI and CFST), the exponent value $w \in [0, 1]$ is a crucial parameter that considerably impacts the performance. The exponent factor can be calculated in different ways which boils down to maximization/minimization of a cost measure. The cost measures include the determinant or the trace of the covariance matrix or Shannon entropy [11] as mentioned in [12]. Further, the exponent factors can be calculated approximately as in [35, 36] rather than the optimization of the nonlinear cost function in order to decrease the computational complexity. The divergence measures such as Kullback-Leibler divergence [11] and Rényi divergence can also be used.
As the exponents of the different sources approach zero or unity, the distinctions between the methods become less noticeable. As a result, the exponents $(w_1, w_2) = (\frac{1}{2}, \frac{1}{2})$ treats the sensor equally in the first fusion. Similarly, the second fusion accepts the output of the first fusion as input with the exponents of $(w_o, w_3) = (\frac{2}{3}, \frac{1}{3})$, i.e., the filter outputs of the sensors have equal importance in terms of the exponent factors.

3.3 Simulation Results

In this section, the fusion algorithms' performance are compared via simulations and real data experiments. The methods used in the experiments are introduced below

- **Kalman filter (KF):** Local Kalman filters are run at the sensor nodes to provide posterior densities to the decentralised fusion algorithms (e.g. CI and CFST). Furthermore, the RMSE values of the local Kalman filters are reported for the performance comparison.

- **Centralized Kalman filter (CKF):** This method processes the measurements sent to the fusion center to obtain posterior density in a centralized manner. If the model parameters are perfectly known, centralized Kalman filter provides theoretically optimal solution. However, scenarios involving parameter mismatches will also be simulated.

- **Covariance intersection (CI):** It is a frequently used method for fusing pos-

teriors in the context of target tracking. Since the underlying input densities are assumed to be Gaussion $p_{x,1}(x) \triangleq \mathcal{N}(x; x_1, P_1)$ and $p_{x,2}(x) \triangleq \mathcal{N}(x; x_2, P_2)$, the mean x_{CI} and the covariance P_{CI} which describe the fused density can be computed according to the following formulas

$$P_{\text{CI}}^{-1} x_{\text{CI}} = w P_1^{-1} x_1 + (1-w) P_2^{-1} x_2, \tag{3.43a}$$

$$P_{\text{CI}}^{-1} = w P_1^{-1} + (1-w) P_2^{-1}. \tag{3.43b}$$

- **Chernoff fusion with student t-distribution (CFST):** It is the proposed algorithm in this thesis, which combines Chernoff fusion with student t-mixture. CFST algorithm consists of four main steps:
  - taking a non-integer power of a student-t distribution;
  - taking a non-integer power of a student-t mixture;
  - multiplying two student-t distributions;
  - merging student-t mixtures.

3.3.1 Scenario Definition

The problem selected for evaluation purposes is based on a single target tracking scenario, in which a target moves on a path simulated according to (3.44a) with the sampling interval $\Delta T = 1\,\text{s}$. A realization of the simulated path is given in Figure 3.6. Meanwhile, local sensors obtain measurements from the moving target according to (3.44b).

The state of the target $x_k = [p_{x,k} \quad p_{y,k} \quad v_{x,k} \quad v_{y,k}]^{\text{T}}$ consists of the planar position and velocity. The state-space model with the additive process and measurement noise assumption is given as

$$x_k = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_{k-1} + \begin{bmatrix} \frac{\Delta T^2}{2} & 0 \\ 0 & \frac{\Delta T^2}{2} \\ \Delta T & 0 \\ 0 & \Delta T \end{bmatrix} w_{k-1}, \tag{3.44a}$$

$$z_{k,m} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x_{k,m} + v_{k,m}, \tag{3.44b}$$

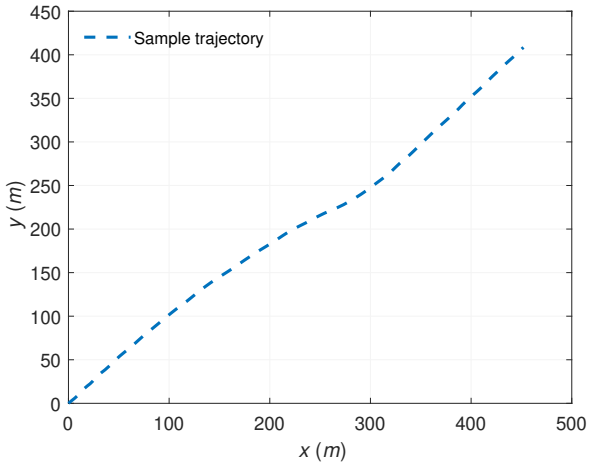


Figure 3.6: Sample trajectory simulated based on (3.44a) with the sampling interval $\Delta T = 1\,\mathrm{s}$.

where m denotes the m^{th} sensor and $w_{k-1} \sim \mathcal{N}(0, Q_{k-1})$ with $Q_{k-1} = \mathrm{diag}(\sigma_w^2, \sigma_w^2)$ and $\sigma_w = 0.5$, $v_{k,m} \sim \mathcal{N}(0, R_{k,m})$ with $R_{k,m} = \mathrm{diag}(\sigma_v^2, \sigma_v^2)$.

In order to assess the performance of the algorithms, root mean square error (RMSE) is used as a performance measure. For M Monte Carlo runs, the average RMSE at time k ($RMSE_k$) is computed using the following formula

$$\text{RMSE}_k = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(\hat{z}_k^i - z_k^i\right)^\top \left(\hat{z}_k^i - z_k^i\right)}, \tag{3.45}$$

where $\hat{z}_k^i$ and z_k^i represent estimated and true portion of the state such as p_k^i and v_k^i at time k in run i, respectively. The average RMSE of the whole scenario over M Monte Carlo runs is calculated as

$$\text{RMSE} = \frac{1}{M}\sum_{i=1}^{M}\sqrt{\frac{1}{T}\sum_{k=1}^{T}\left(\hat{z}_k^i - z_k^i\right)^\top \left(\hat{z}_k^i - z_k^i\right)}. \tag{3.46}$$

For CFST algorithm, the degrees of freedom ν is another essential parameter. The non-integer power of a student t-distribution or student t-mixture causes a decrease in the degrees of freedom, which depends on the exponent value. Further, $\nu > 2$ is a requirement for the algorithm's feasibility, which results in preventing the degrees of freedom ν from falling below a certain value. Through all simulations, initial degrees of freedom is selected as $\nu_0 = 10$.

In order to demonstrate the performance of the algorithms, three distinct scenarios are constructed: the case with known sensor characteristics, the case with one sensor failure, and the last one is the case of one sensor failure with bias.

3.3.2 Case 1: Known Sensor Characteristics

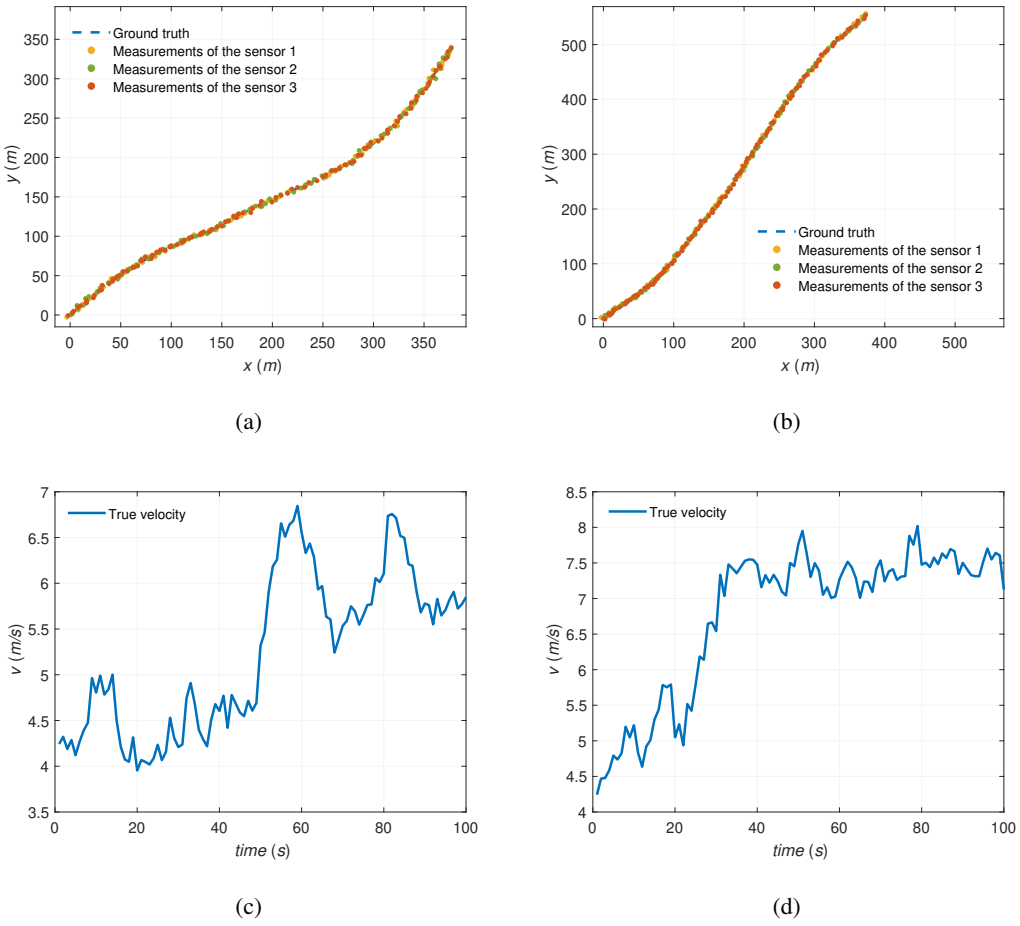
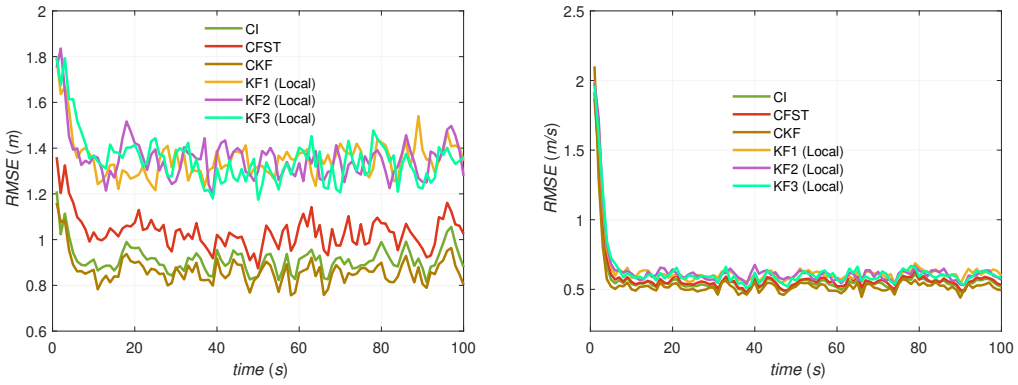


Figure 3.7: (a) and (b) show the realizations of the sensor measurements and the ground truth. (c) and (d) show the corresponding true velocity values of the realizations.

In the first scenario, a single target is tracked by using the measurements of three sensors. It is assumed that the algorithms precisely know sensor characteristics, i.e., the measurement noise variances are known exactly by KF, CKF, CI and CFST algorithms. Further, measurement noise standard deviation is chosen as $\sigma_v = 1.4$ for



(a) (b)

Figure 3.8: (a) (left figure) shows the average position RMSE of the algorithms at each time instant over 100 Monte Carlo runs for Case 1. (b) (right figure) shows the average velocity RMSE of the algorithms at each time instant over 100 Monte Carlo runs for Case 1.

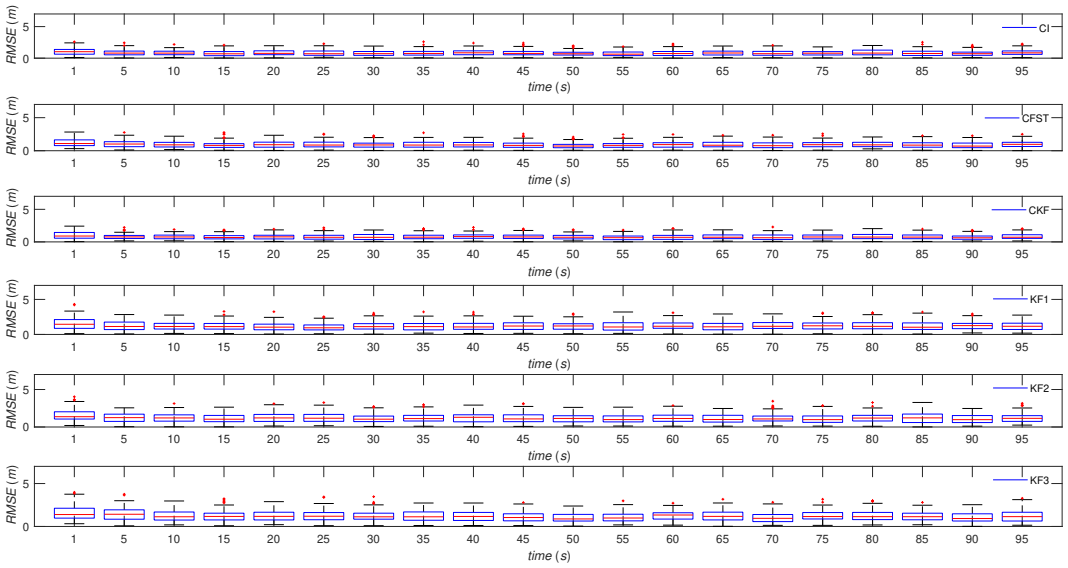
all three sensors. For this case, Figure 3.7(a) and Figure 3.7(b) illustrate the realization of the sensor measurements and the ground truth of the target trajectory. Figure 3.7(c) and Figure 3.7(d) are the corresponding true velocity values of the realizations in Figure 3.7(a) and Figure 3.7(b), respectively.

The average RMSE values of the scenario in Case 1 are given in Table 3.1. The

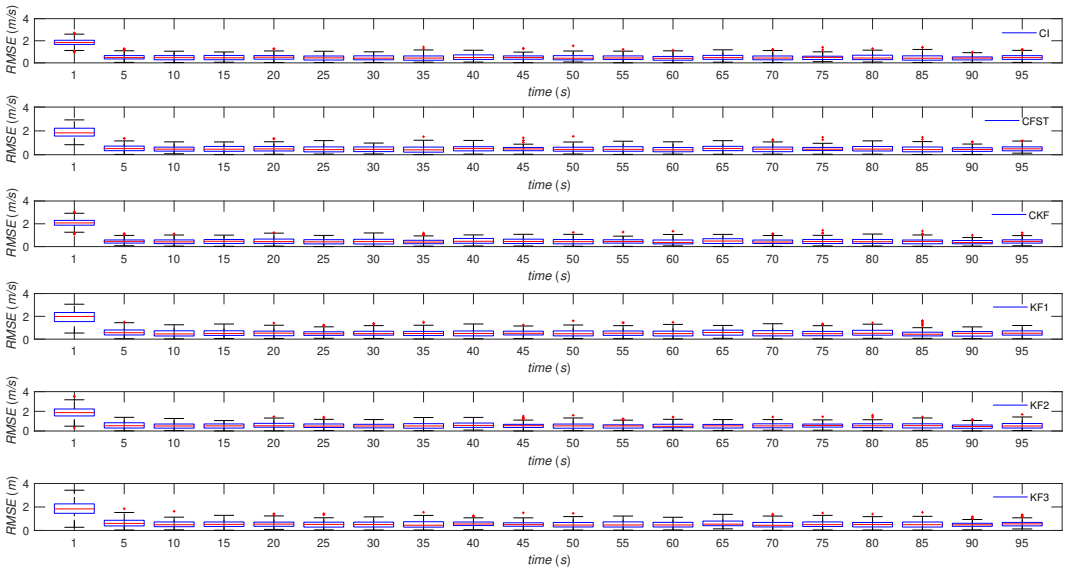
Table 3.1: Average RMSE values of the position and velocity for the Case 1.

	RMSE (*m*)	**RMSE** (*m/s*)
CI	0.91	0.59
CKF	0.86	0.57
KF1	1.35	0.66
KF2	1.36	0.66
KF3	1.35	0.64
CFST	1.03	0.61

average (over 100 Monte Carlo runs) position and velocity RMSE values calculated using (3.45) are depicted in Figure 3.8. Figure 3.9 shows the box and whisker plots for the average position and velocity RMSE values of the algorithms.



(a)



(b)

Figure 3.9: (a) (top figure) shows the box and whisker plot for average position RMSE of the algorithms at each time instant over 100 Monte Carlo runs for Case 1. (b) (bottom figure) shows the box and whisker plot for average velocity RMSE of the algorithms at each time instant over 100 Monte Carlo runs for Case 1.

The following observations can be extracted according to the results in Table 3.1 and Figures 3.8 and 3.9.

- The fusion algorithms perform better than local KFs.
- CKF outperforms all algorithms since its optimality condition is satisfied.
- Under the known parameters and valid Gaussian assumptions, CI performs better than CFST.
- According to the box and whisker plots, the RMSE value spreads of the algorithms at the same instances are comparable and narrower, which indicates that the average RMSE values at each instant are accurate depictions of the error and that there is no algorithmic advantage in terms of the RMSE value spread.

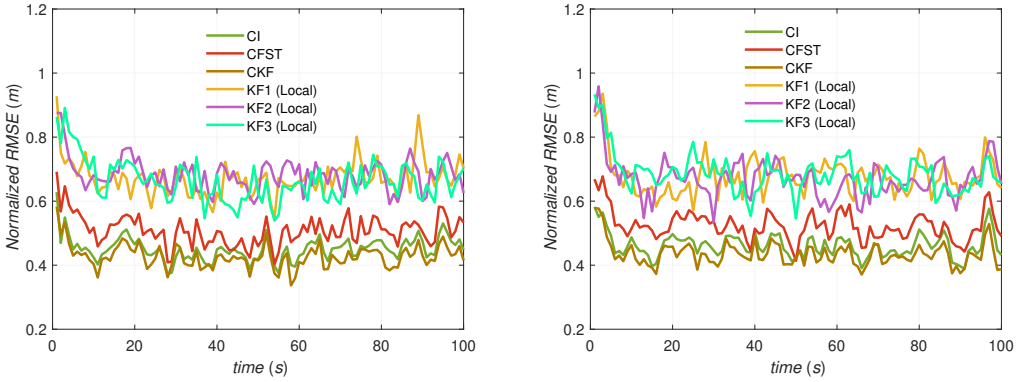


Figure 3.10: (a) (left figure) shows the normalized average position RMSE in the x direction of the algorithms at each time instant over 100 Monte Carlo runs for Case 1. (b) (right figure) shows the normalized average position RMSE in the y direction of the algorithms at each time instant over 100 Monte Carlo runs for Case 1.

Figure 3.10 shows the normalized average position RMSE in x and y direction of the algorithms at each time instant over 100 Monte Carlo runs for Case 1. RMSE values at each time instant are normalized by the measurement noise standard deviation $\sigma_v = 1.4$. Figure 3.10 indicates that all methods perform better than using the measurements directly for the position estimation.

3.3.3 Case 2: One Sensor Failure

In this case, it is assumed that one of the sensors fails to produce the measurements with the expected quality due to an unknown failure. The measurement noise standard deviation of this sensor is chosen as $\sigma_v = 7$ instead of $\sigma_v = 1.4$. Figure 3.11(a) and Figure 3.11(b) show the realizations of the sensor measurements and the ground truth of the target trajectory. Figure 3.7(c) and Figure 3.7(d) are the corresponding true velocity values of the realizations in Figure 3.7(a) and Figure 3.7(b), respectively.

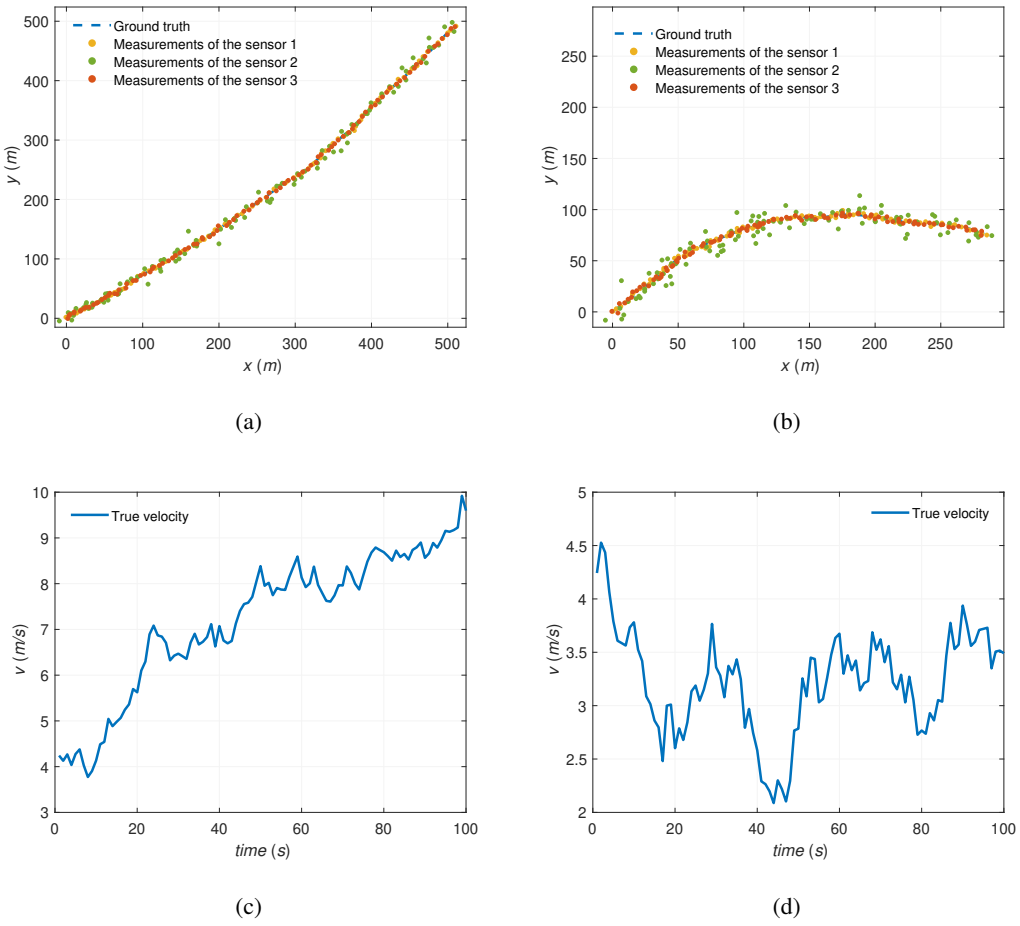
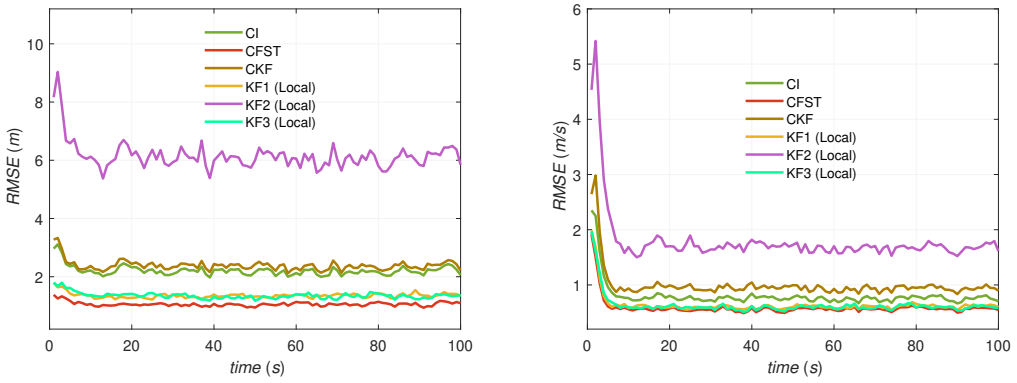


Figure 3.11: (a) and (b) show the realizations of the sensor measurements and the ground truth. (c) and (d) show the corresponding true velocity values of the realizations.



(a) (b)

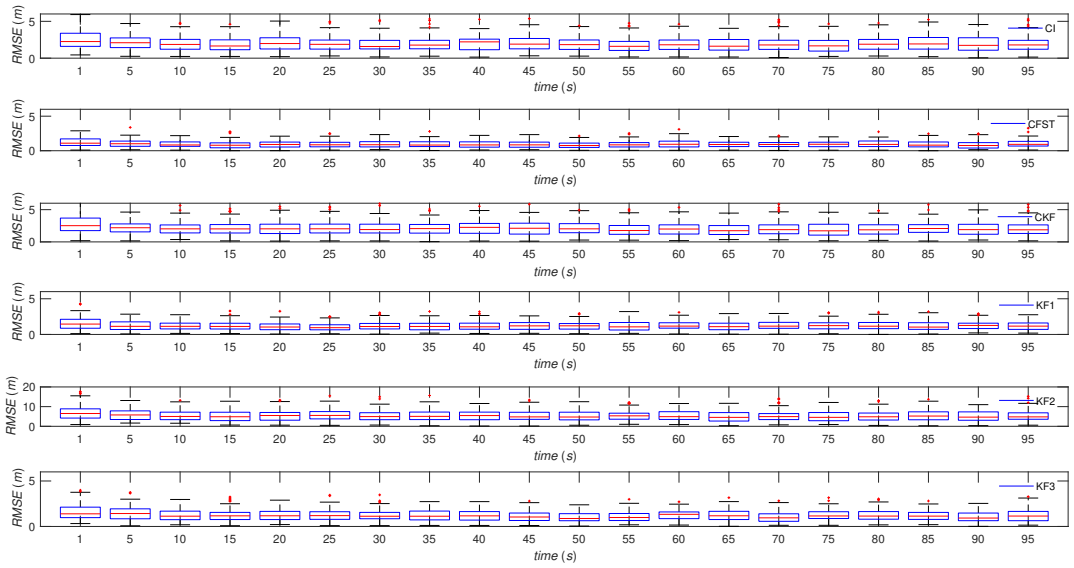
Figure 3.12: (a) (left figure) shows the average position RMSE of the algorithms at each time instant over 100 Monte Carlo runs for Case 2. (b) (right figure) shows the average velocity RMSE of the algorithms at each time instant over 100 Monte Carlo runs for Case 2.

Table 3.2: Average RMSE values of the position and velocity for the Case 2.

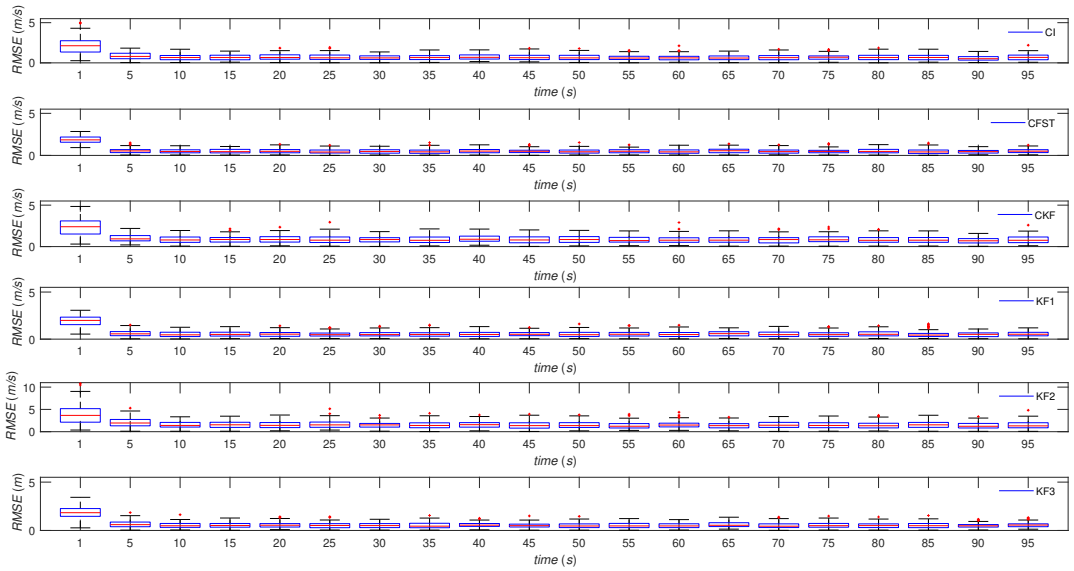
	RMSE (*m*)	**RMSE** (*m/s*)
CI	2.21	0.83
CKF	2.39	1.03
KF1	1.35	0.66
KF2	6.15	1.87
KF3	1.35	0.65
CFST	1.06	0.61

The average RMSE values of the scenario in Case 2 are given in Table 3.2. The average (over 100 Monte Carlo runs) position and velocity RMSE values calculated using (3.45) are depicted in Figure 3.12. Based on the results in Table 3.2 and Figure 3.12, CFST performs better than the other fusion algorithms and KFs. Furthermore, in such a case, the Gaussian assumption based methods such as CI and CKF fail in fusion performance. These results can be interpreted as the robustness of CFST algorithm in case of the model parameter mismatches.

Figure 3.13(a) and Figure 3.13(b) show the box and whisker plots for the average



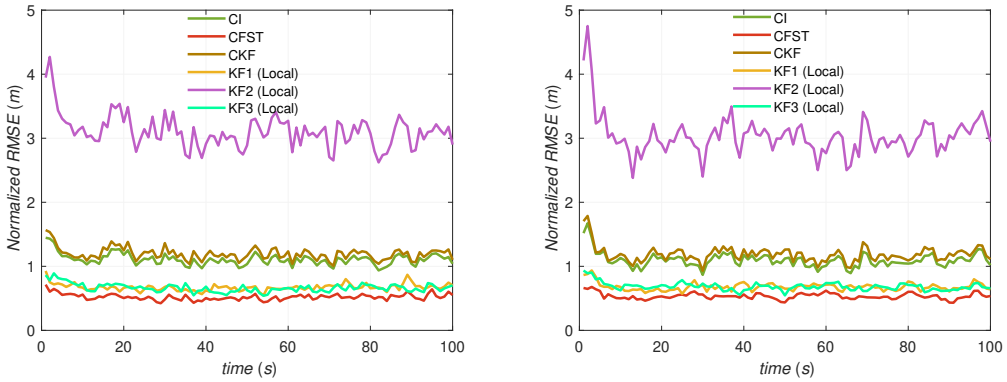
(a)



(b)

Figure 3.13: (a) (top figure) shows the box and whisker plot for average position RMSE of the algorithms at each time instant over 100 Monte Carlo runs for Case 2. (b) (bottom figure) shows the box and whisker plot for average velocity RMSE of the algorithms at each time instant over 100 Monte Carlo runs for Case 2.

position and velocity RMSE values of the algorithms, respectively. According to the results in Figure 3.13, RMSE value spreads of the algorithms at the same instances are very similar and narrower, which means that the average RMSE values at each instant are good representations of the error, and there is no superiority between the algorithms in terms of the RMSE values spread.



(a) (b)

Figure 3.14: (a) (left figure) shows the normalized average position RMSE in the x direction of the algorithms at each time instant over 100 Monte Carlo runs for Case 2. (b) (right figure) shows the normalized average position RMSE in the y direction of the algorithms at each time instant over 100 Monte Carlo runs for Case 2.

Figure 3.14 shows the normalized average position RMSE in x and y direction of the algorithms at each time instant over 100 Monte Carlo runs for Case 2. RMSE values at each time instant are normalized by the measurement noise standard deviation $\sigma_v = 1.4$ (measurement noise standard deviation of the sensor with no failure). Based on the results in Figure 3.14, CFST and local Kalman filters without sensor failure perform better than using the measurements directly for the position estimation. On the other hand, using the measurements of the sensors without failure directly is more advantageous than CI and CKF in terms of the position RMSE.

3.3.4 Investigating Less Frequent Sensor Failure

This case aims to observe how the performance of the algorithms in Case 2 changes as the error production frequency of the sensor with failure decreases. Once per four successive instants, the measurement noise standard deviation of this sensor is set to $\sigma_v = 7$ instead of $\sigma_v = 1.4$. Figure 3.15(a) and Figure 3.15(b) show the realizations of the sensor measurements and the ground truth of the target trajectory. Figure 3.15(c) and Figure 3.15(d) are the corresponding true velocity values of the realizations in Figure 3.15(a) and Figure 3.15(b), respectively.

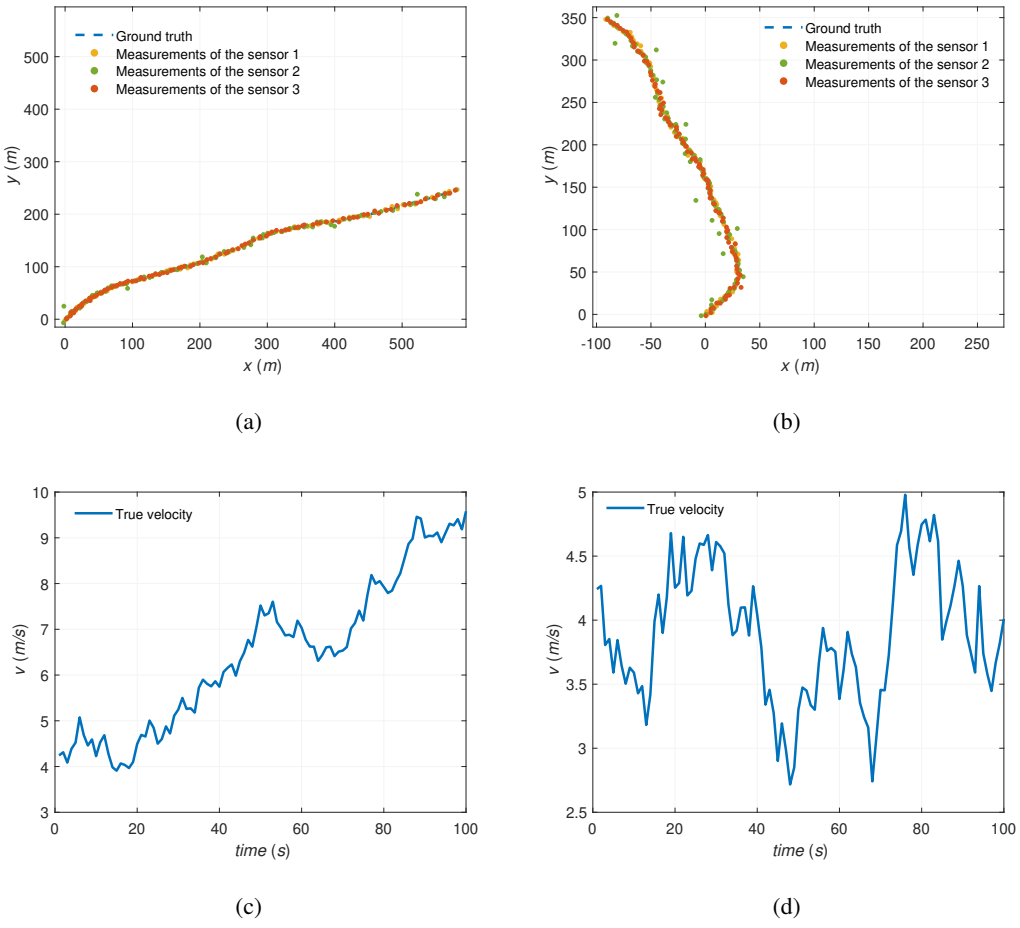
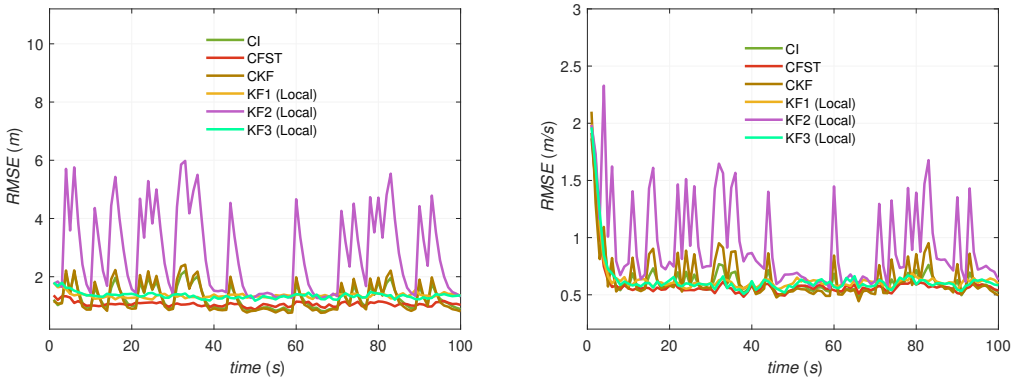


Figure 3.15: (a) and (b) show the realizations of the sensor measurements and the ground truth. (c) and (d) show the corresponding true velocity values of the realizations.



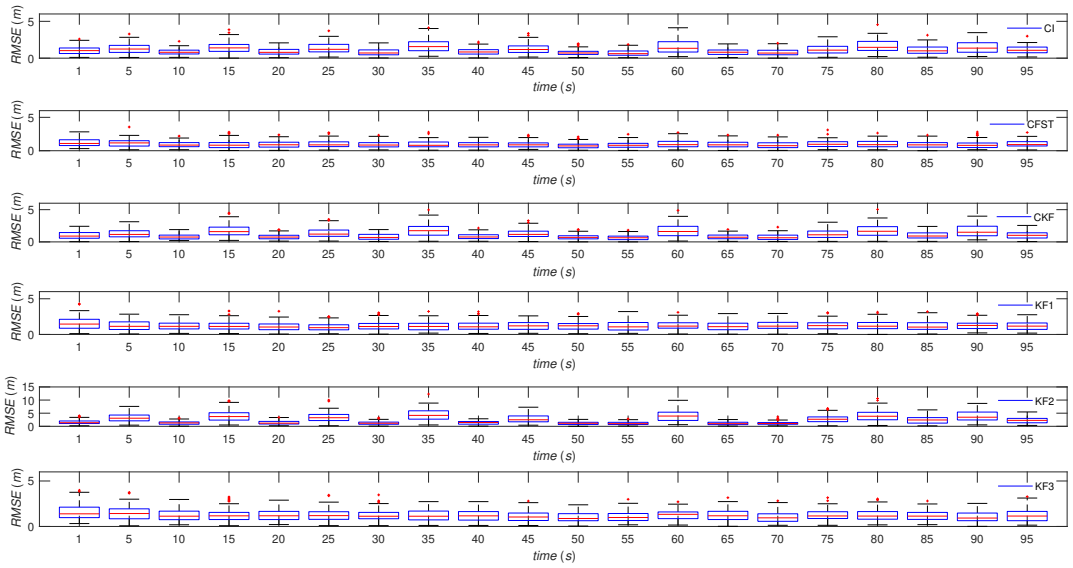
(a) (b)

Figure 3.16: (a) (left figure) shows the average position RMSE of the algorithms at each time instant over 100 Monte Carlo runs for investigating less frequent sensor failure. (b) (right figure) shows the average velocity RMSE of the algorithms at each time instant over 100 Monte Carlo runs for investigating less frequent sensor failure.

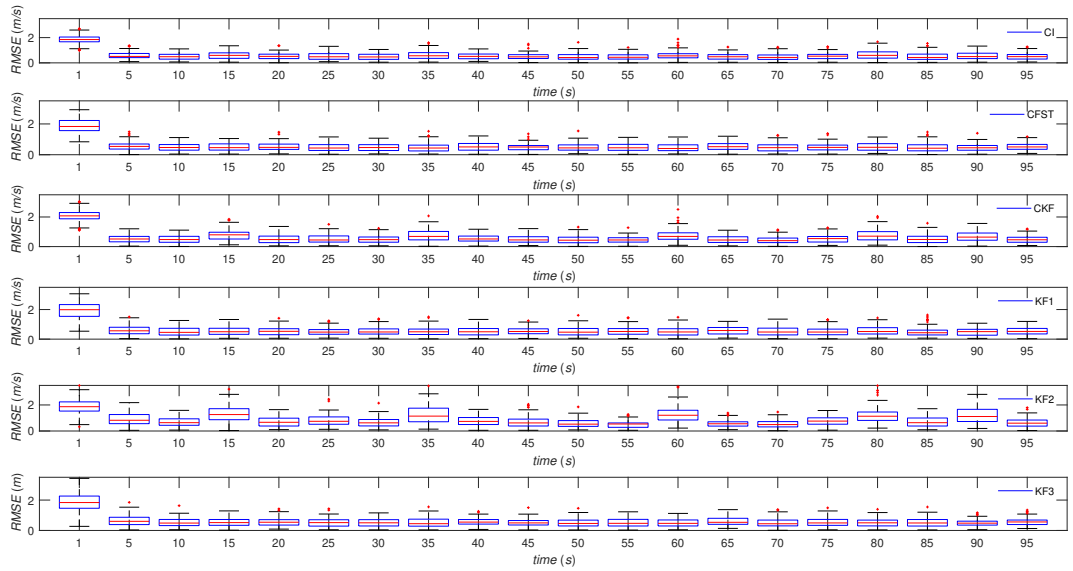
Table 3.3: Average RMSE values of the position and velocity for investigating less frequent sensor failure.

	RMSE (*m*)	**RMSE** (*m/s*)
CI	1.32	0.64
CKF	1.35	0.68
KF1	1.35	0.66
KF2	3.12	1.00
KF3	1.35	0.65
CFST	1.06	0.61

The average RMSE values of the scenario for investigating less frequent sensor failure are given in Table 3.3. The average (over 100 Monte Carlo runs) position and velocity RMSE values calculated using (3.45) are depicted in Figure 3.16. Based on the results in Table 3.3 and Figure 3.16, CFST outperforms the other fusion algorithms and KFs. Compared to the results in Case 2, the performance of CFST is less affected by the decrease in the sensor's error production frequency than other fusion algorithms, such as CI and CKF.



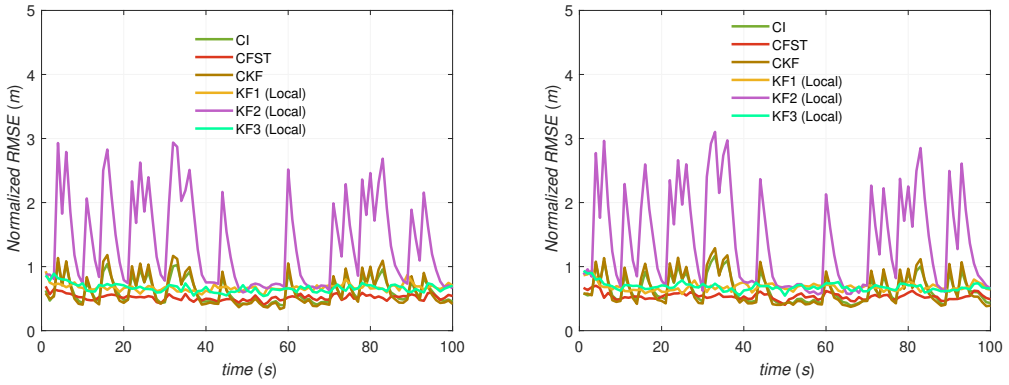
(a)



(b)

Figure 3.17: (a) (top figure) shows the box and whisker plot for average position RMSE of the algorithms at each time instant over 100 Monte Carlo runs for investigating less frequent sensor failure. (b) (bottom figure) shows the box and whisker plot for average velocity RMSE of the algorithms at each time instant over 100 Monte Carlo runs for investigating less frequent sensor failure.

Figure 3.17(a) and Figure 3.17(b) show the box and whisker plots for the average position and velocity RMSE values of the algorithms, respectively. According to the results in Figure 3.13, the algorithms' RMSE value spreads at the same instances are highly comparable and small, indicating that the average RMSE values at each instant are accurate depictions of the error and that no method performs better than another method in terms of RMSE value spread.



(a) (b)

Figure 3.18: (a) (left figure) shows the normalized average position RMSE in the x direction of the algorithms at each time instant over 100 Monte Carlo runs for investigating less frequent sensor failure. (b) (right figure) shows the normalized average position RMSE in the y direction of the algorithms at each time instant over 100 Monte Carlo runs for investigating less frequent sensor failure.

Figure 3.18 shows the normalized average position RMSE in x and y direction of the algorithms at each time instant over 100 Monte Carlo runs for investigating less frequent sensor failure. RMSE values at each time instant are normalized by the measurement noise standard deviation $\sigma_v = 1.4$ (measurement noise standard deviation of the sensor with no failure). Based on the results in Figure 3.18, CFST and local Kalman filters without sensor failure is more advantageous than using the measurements directly for the position estimation as in Case 2. On the other hand, using the sensors' measurements without failure directly performs better than CI and CKF. When all sensors produce reasonable measurements, CI and CKF are better than using the measurements.

3.3.5 Case 3: One Sensor With Bias

In this case, it is assumed that one of the sensors generate measurements with bias in each dimension ($10\,\mathrm{m}$) in addition to the measurement noise standard deviation with $\sigma_v = 1.4$. For this case, Figure 3.19(a) and Figure 3.19(b) illustrate the realization of the sensor measurements and the ground truth of the target trajectory. Figure 3.19(c) and Figure 3.19(d) are the corresponding true velocity values of the realizations in Figure 3.19(a) and Figure 3.19(b), respectively.

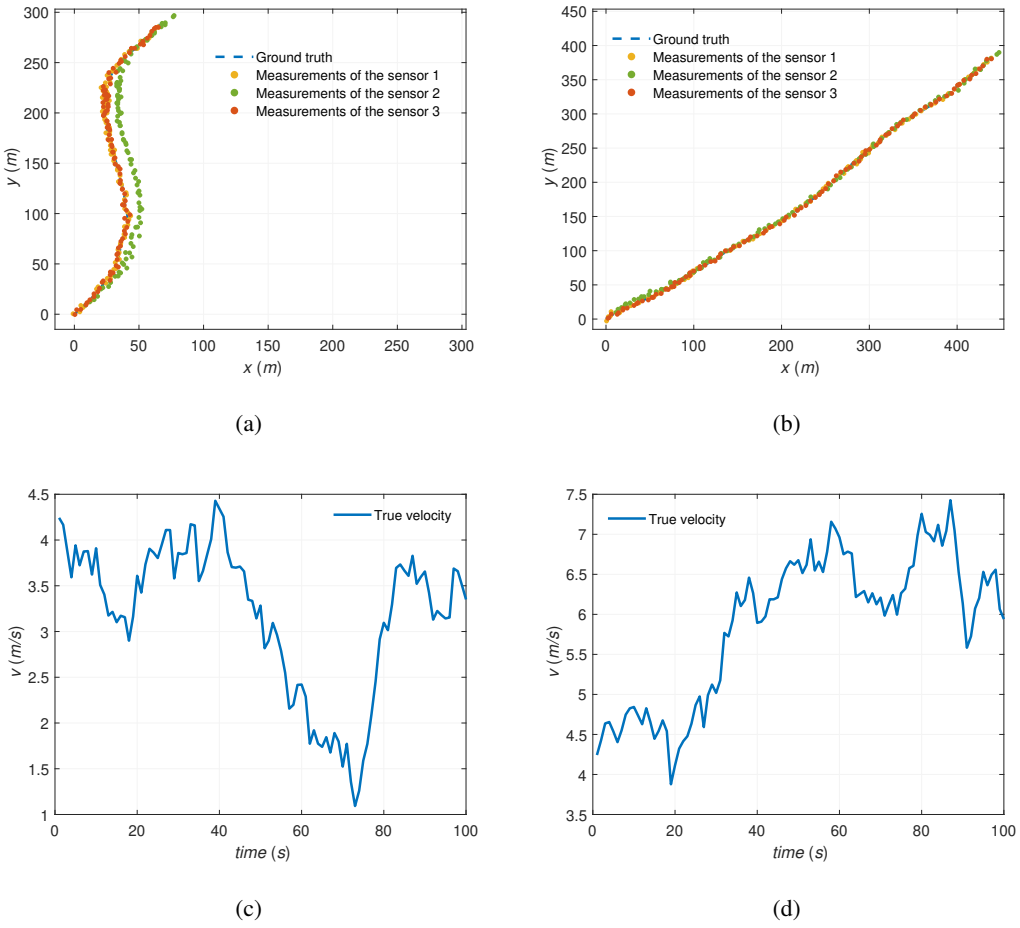
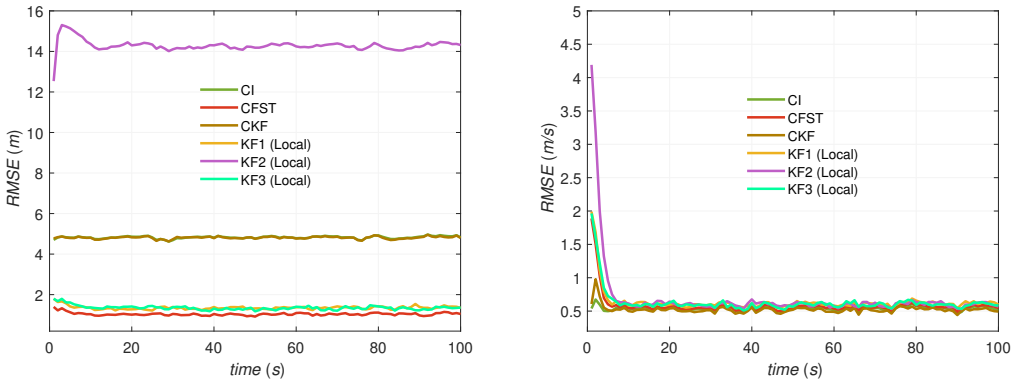


Figure 3.19: (a) and (b) show the realizations of the sensor measurements and the ground truth. (c) and (d) show the corresponding true velocity values of the realizations.

The average RMSE values of the scenario in Case 3 are given in Table 3.4. The average (over 100 Monte Carlo runs) position and velocity RMSE values calculated using



(a) (b)

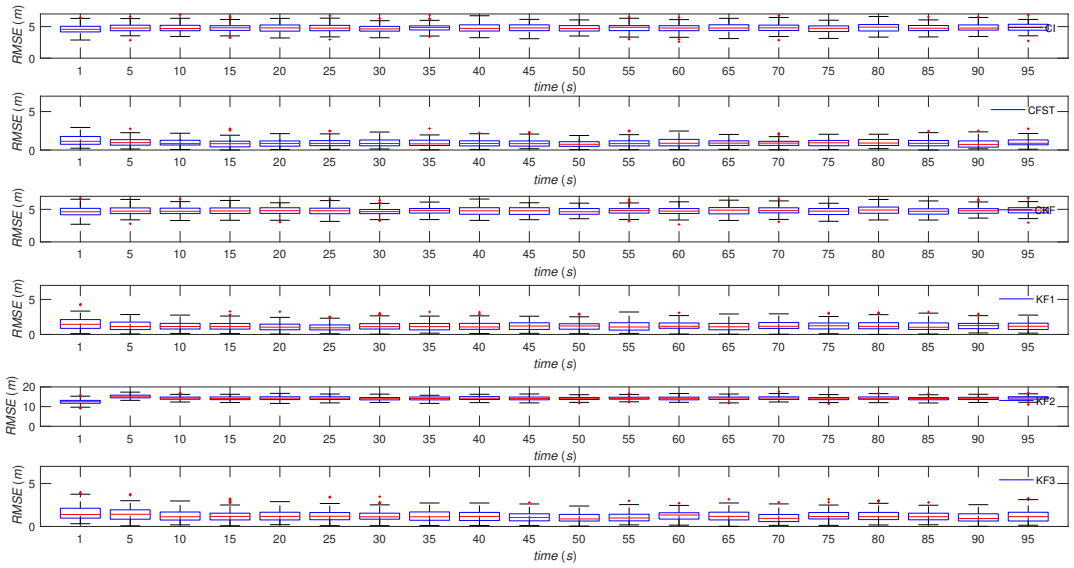
Figure 3.20: (a) (left figure) shows the average position RMSE of the algorithms at each time instant over 100 Monte Carlo runs for Case 3. (b) (right figure) shows the average velocity RMSE of the algorithms at each time instant over 100 Monte Carlo runs for Case 3.

Table 3.4: Average RMSE values of the position and velocity for the Case 3 ($10\,\mathrm{m}$ in each dimension).

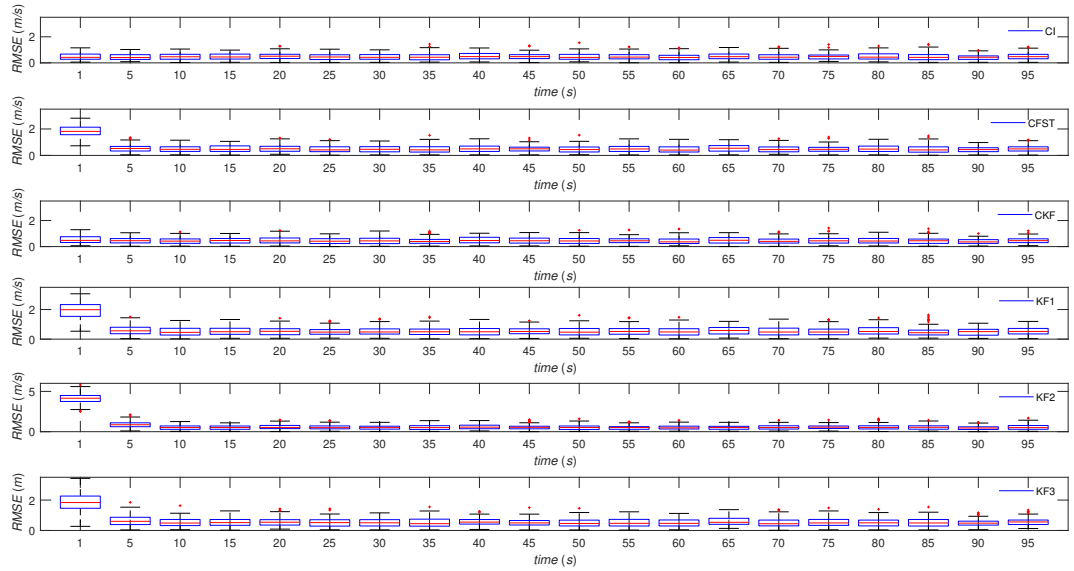
	RMSE (*m*)	**RMSE** (*m/s*)
CI	4.82	0.54
CKF	4.81	0.52
KF1	1.35	0.66
KF2	14.29	0.83
KF3	1.35	0.65
CFST	1.04	0.63

(3.45) are given in Figure 3.20. Based on the results in Table 3.4 and Figure 3.20, CFST outperforms other fusion algorithms and KFs, which is related to the robustness as in Case 2 although the sensor error is increased.

Figure 3.21(a) and Figure 3.21(b) show the box and whisker plots for the average position and velocity RMSE values of the algorithms, respectively. Figure 3.21 shows that the average RMSE values at each instant are good representations of the error, and there is no superiority between the algorithms in terms of the RMSE value spread,



(a)

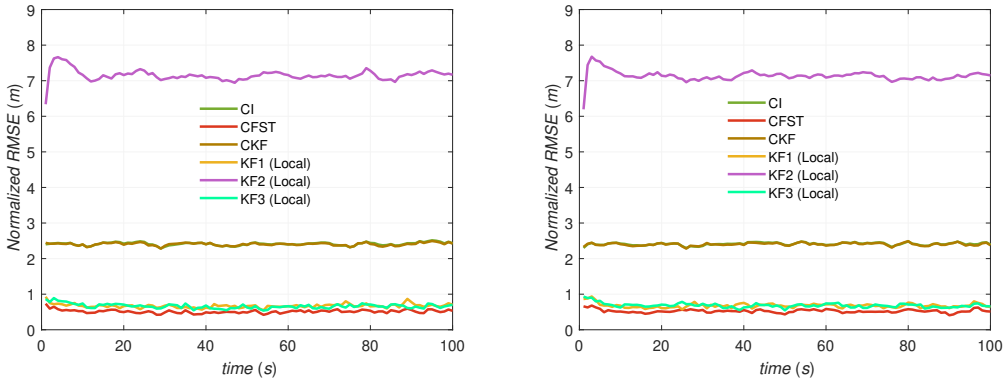


(b)

Figure 3.21: (a) (top figure) shows the box and whisker plot for average position RMSE of the algorithms at each time instant over 100 Monte Carlo runs for Case 3. (b) (bottom figure) shows the box and whisker plot for average velocity RMSE of the algorithms at each time instant over 100 Monte Carlo runs for Case 3.

as shown by the algorithms' extremely similar and shorter RMSE value spreads at the same instances.

Figure 3.22 shows the normalized average position RMSE in x and y direction of



(a) (b)

Figure 3.22: (a) (left figure) shows the normalized average position RMSE in the x direction of the algorithms at each time instant over 100 Monte Carlo runs for Case 3. (b) (right figure) shows the normalized average position RMSE in the y direction of the algorithms at each time instant over 100 Monte Carlo runs for Case 3.

the algorithms at each time instant over 100 Monte Carlo runs for Case 3. RMSE values at each time instant are normalized by the measurement noise standard deviation $\sigma_v = 1.4$ (measurement noise standard deviation of the sensor with no failure). Based on the results in Figure 3.22, instead of using the readings directly for position estimation, CFST and local Kalman filters without sensor failure are preferable as in Case 3. On the other hand, using the sensors' measurements without failure directly performs better than fusion methods based on Gaussian assumption such as CI and CKF.

3.3.6 One Dimensional Analysis For CFST

In this subsection, the aim is to investigate the steps of CFST algorithm with one dimensional example. The experiments are conducted for CFST and the numerically calculated Chernoff fusion with student t-distribution in a lower dimensional state-space model. The problem considers a stationary object and noisy measurements that

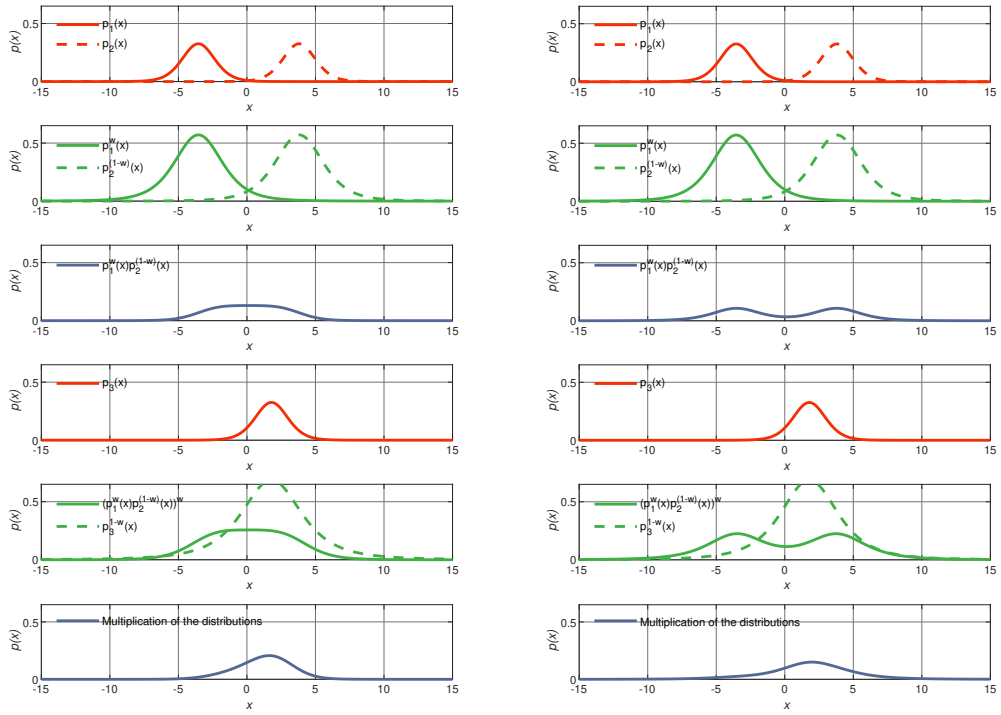
are collected from three different sensors with the sampling interval $\Delta T = 1\,\mathrm{s}$. The state of the target $x_k = p_{x,k}$ is described only with the one-dimensional position and the measurement model is as follows

$$z_{k,m} = x_k + v_{k,m}, \tag{3.47}$$

where m denotes the m^{th} sensor and $v_{k,m} \sim \mathcal{N}(0, R_{k,m})$ with $R_{k,m} = \mathrm{diag}(\sigma_v^2, \sigma_v^2)$. The actual position of the target is specified as $x_k = 0$ for the simulations. The figures below illustrate outputs of the three critical steps of CFST as follows

- 1$^{\text{st}}$ row of the figures shows inputs distributed with student t-distribution, i.e., $p_1(x)$ and $p_2(x)$.
- 2$^{\text{nd}}$ row of the figures shows the non-integer power of student t-distributions, i.e., $p_1^{(1-w)}(x)$ and $p_2^w(x)$.
- 3$^{\text{rd}}$ row of the figures shows the multiplication of the distributions in the second row, i.e., $p_1^{(1-w)}(x)p_2^w(x)$.
- 4$^{\text{th}}$ row of the figures shows input distributed with student t-distribution, i.e., $p_3(x)$.
- 5$^{\text{th}}$ row of the figures shows the non-integer power of the student t-mixture in the third row, i.e., $(p_1^{(1-w)}(x)p_2^w(x))^{(1-w)}$ and the non-integer power of the student t-distribution in the forth row, i.e., $p_3^w(x)$
- 6$^{\text{th}}$ row of the figures shows the multiplication of the student t-mixture and student t-distribution in the fifth row, i.e., $(p_1^{(1-w)}(x)p_2^w(x))^{1-w}p_3^w(x)$.

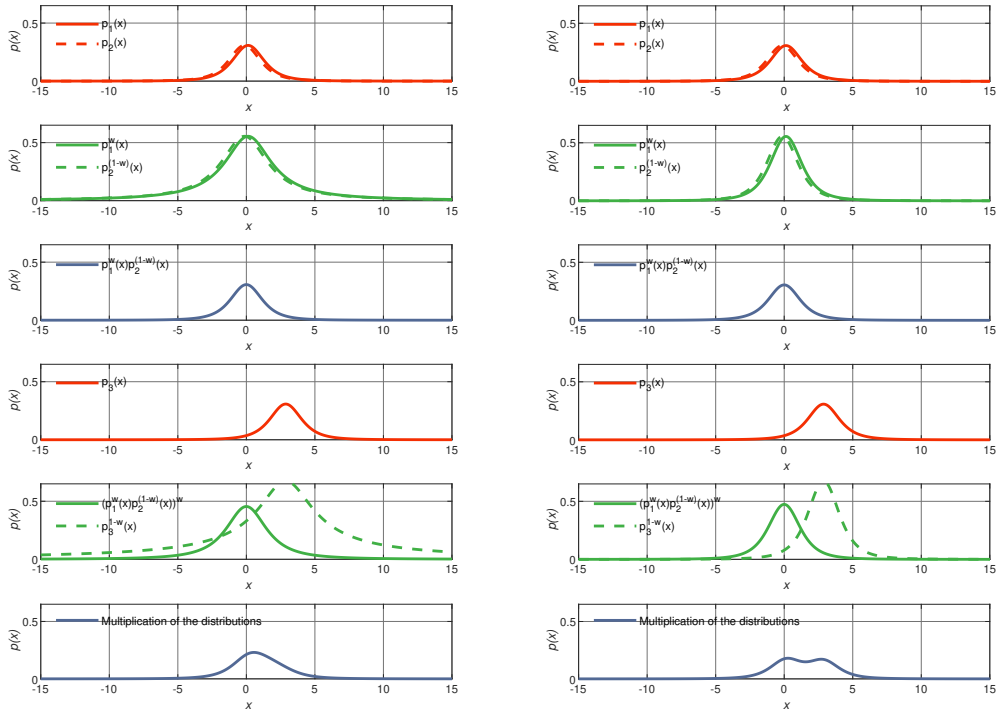
Furthermore, the numerically calculated counterparts of these steps are also available in the figures. The initial degrees of freedom is accepted as $\nu = 10$ for Figures 3.23 and 3.24. For the results in Figure 3.25, initial degrees of freedom is taken as $\nu = 3$ and the scenario is the same as the one in Figure 3.23. Based on the results in Figures 3.23 and 3.25, the non-integer power of $p_1(x)$, $p_2(x)$ and $p_3(x)$ as a result of CFST and numerical computation are completely same for Figure 3.23 since there is no approximation in this step for CFST. On the other hand, when initial degrees of freedom is taken as $\nu = 3$, the non-integer power of the distributions as a result of CFST and



(a) (b)

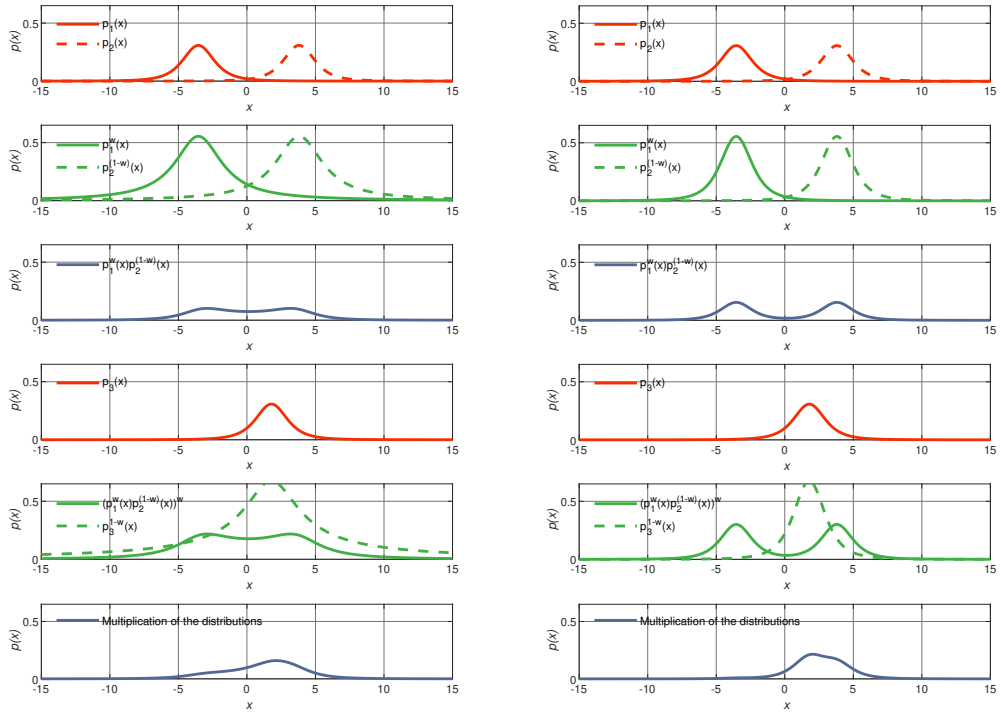
Figure 3.23: (a) (left figure) shows the numerically calculated Chernoff fusion with student t-distributions outputs. (b) (right figure) shows CFST outputs for each step.

numerical computation start to separate. It is caused by preventing the degrees of freedom from falling below a certain value because covariance is available if $\nu > 2$. Apart from these, the multiplication of the student t-distributions is the step where the difference between CFST and the numerical computation is observed the most. Approximating multiplication as the sum of the student t-distributions highlights the multi-modal structure comparing numerical computation.



(a) (b)

Figure 3.24: (a) (left figure) shows the numerically calculated Chernoff fusion with student t-distributions outputs. (b) (right figure) shows CFST outputs for each step.



(a) (b)

Figure 3.25: (a) (left figure) shows the numerically calculated Chernoff fusion with student t-distributions outputs. (b) (right figure) shows CFST outputs for each step.

3.3.7 Discussion: CFST with memory

This part examines the alternative of CFST in terms of additional input. Suppose the fusion output of the previous time instant is provided to CFST algorithm after the time update. In that case, the method is called CFST with memory to distinguish the two approaches. The problem selected for evaluation purposes is the same as in Section 3.3.1. Again, three different experiments with the match and mismatch between the one sensor's actual and known measurement noise standard deviation are realized to compare the algorithms. The pseudo-code of the CFST with memory algorithm for a specific time instant is provided in Algorithm 1.

Algorithm 1 CFST with memory

Input: $\{p_i(x)\}_{i=1}^{N}$ where $p_i(x) \sim t_{\nu_i}(x; \mu_i, \Sigma_i)$ with the exponent factor w_i and $p_o(x)$ with the exponent factor w_o

Output: $q_o(x)$

for $i = 1 : N$ **do**

Compute $p_o^{w_o}(x)$ and $p_i^{w_i}(x)$

Compute $p_o^{w_o}(x)p_i^{w_i}(x)$

$p_o(x) \leftarrow Normalize\ p_o^{w_o}(x)p_i^{w_i}(x)$

end for

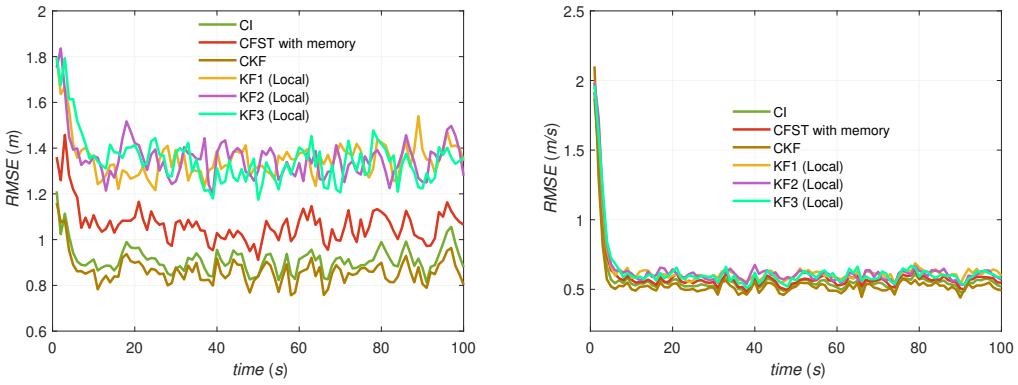
$p_o(x) \leftarrow TimeUpdate(p_o(x))$

$q_o(x) \leftarrow Merge(p_o(x))$

3.3.7.1 Case 1: Known Sensor Characteristics

The measurements of three sensors are used to track a single target in this scenario. It is expected that the algorithms perfectly know sensor characteristics, i.e., the KF, CKF, CI, and CFST algorithms precisely know measurement noise variances. Furthermore, for all three sensors, the measurement noise standard deviation is set to $\sigma_v = 1.4$.

The average RMSE values of the scenario in Case 1 are given in Table 3.5. The average (over 100 Monte Carlo runs) position and velocity RMSE values calculated



(a) (b)

Figure 3.26: (a) (left figure) shows the average position RMSE of the algorithms at each time instant over 100 Monte Carlo runs for Case 1. (b) (right figure) shows the average velocity RMSE of the algorithms at each time instant over 100 Monte Carlo runs for Case 1.

Table 3.5: Average RMSE values of the position and velocity for the Case 1.

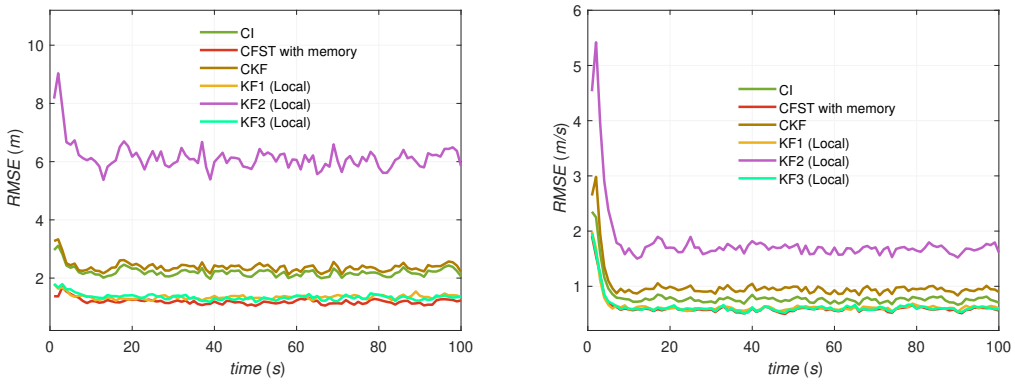
	RMSE (*m*)	**RMSE** (*m/s*)
CI	0.91	0.59
CKF	0.86	0.57
KF1	1.35	0.66
KF2	1.36	0.64
KF3	1.06	0.62
CFST with memory	1.35	2.18

using (3.45) are depicted in Figure 3.26.

According to the results in Table 3.5 and Figure 3.8, the fusion algorithms perform better than local KFs. Further, the fusion algorithms that assume underlying input densities as Gaussian perform better than CFST with memory.

3.3.7.2 Case 2: One Sensor Failure

In this case, one of the sensors fails to produce measurements consistent with the known measurement noise standard deviation. This sensor's measurement noise standard deviation is set to $\sigma_v = 7$ rather than $\sigma_v = 1.4$.



(a) (b)

Figure 3.27: (a) (left figure) shows the average position RMSE of the algorithms at each time instant over 100 Monte Carlo runs for Case 2. (b) (right figure) shows the average velocity RMSE of the algorithms at each time instant over 100 Monte Carlo runs for Case 2.

Table 3.6: Average RMSE values of the position and velocity for the Case 2.

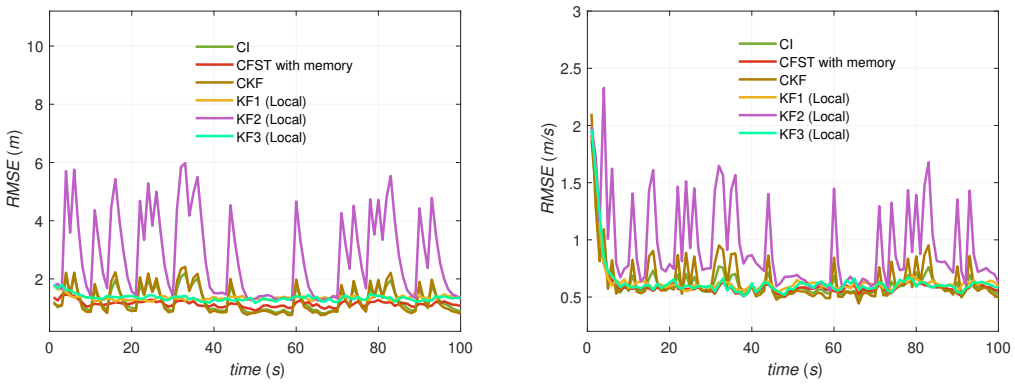
	RMSE (*m*)	**RMSE** (*m/s*)
CI	2.21	0.83
CKF	2.39	1.03
KF1	1.35	0.66
KF2	6.15	1.87
KF3	1.35	0.65
CFST with memory	1.19	0.64

The average RMSE values of the scenario in Case 2 are given in Table 3.6. The average (over 100 Monte Carlo runs) position and velocity RMSE values calculated using (3.45) are depicted in Figure 3.27. According to the results in Table 3.6 and Figure

3.27, the fusion methods that assumes underlying input densities as Gaussian fail in fusion performance in case of the model parameter mismatches. On the other hand, CFST with memory outperforms fusion algorithms and KFs in case of the model parameter mismatches. This indicates the robustness of CFST with memory algorithm in case of outlier.

3.3.7.3 Investigating Less Frequent Sensor Failure

This section examines how the algorithms' performance changes when the sensor's error production frequency decreases. The measurement noise standard deviation of this sensor is set to $\sigma_v = 7$ instead of $\sigma_v = 1.4$ once every four instants. The



(a) (b)

Figure 3.28: (a) (left figure) shows the average position RMSE of the algorithms at each time instant over 100 Monte Carlo runs for investigating less frequent sensor failure. (b) (right figure) shows the average velocity RMSE of the algorithms at each time instant over 100 Monte Carlo runs for investigating less frequent sensor failure.

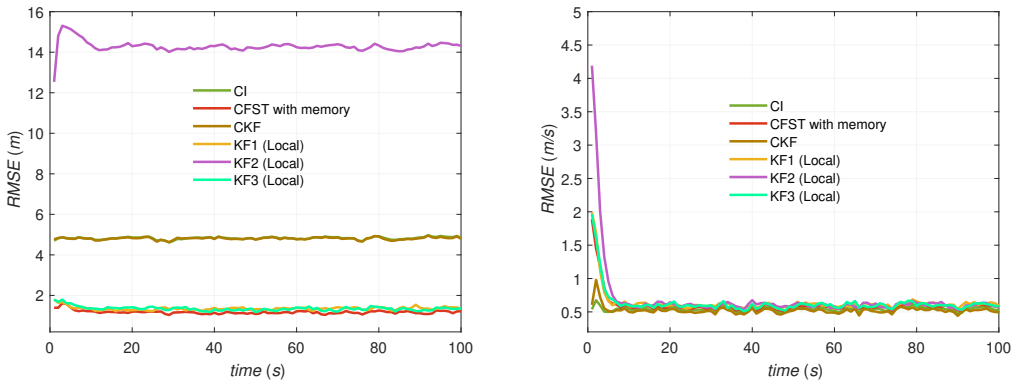
average RMSE values of the scenario for investigating less frequent sensor failure are given in Table 3.7. The average (over 100 Monte Carlo runs) position and velocity RMSE values calculated using (3.45) are depicted in Figure 3.28. Based on the results in Table 3.7 and Figure 3.28, CFST performs better than the other fusion algorithms and KFs despite the decrease in the sensor's error production frequency. Furthermore, among the fusion algorithms, CFST with memory is the least affected by the sensor's error production frequency.

Table 3.7: Average RMSE values of the position and velocity for for investigating less frequent sensor failure.

	RMSE (*m*)	**RMSE** (*m/s*)
CI	1.32	0.64
CKF	1.35	0.68
KF1	1.35	0.66
KF2	3.12	1.01
KF3	1.35	0.65
CFST with memory	1.14	0.63

3.3.7.4 Case 3: One Sensor With Bias

In this case, it is assumed that one of the sensors generate measurements with bias in each dimension $(10\,\mathrm{m})$ in addition to the measurement noise standard deviation with $\sigma_v = 1.4$.



(a) (b)

Figure 3.29: (a) (left figure) shows the average position RMSE of the algorithms at each time instant over 100 Monte Carlo runs for Case 3. (b) (right figure) shows the average velocity RMSE of the algorithms at each time instant over 100 Monte Carlo runs for Case 3.

The average RMSE values of the scenario in Case 3 are given in Table 3.8. The average (over 100 Monte Carlo runs) position and velocity RMSE values calculated using

Table 3.8: Average RMSE values of the position and velocity for the Case 3 ($10\,\mathrm{m}$ in each dimension).

	RMSE (*m*)	**RMSE** (*m/s*)
CI	4.82	0.54
CKF	4.81	0.52
KF1	1.35	0.66
KF2	14.29	0.83
KF3	1.35	0.65
CFST with memory	1.17	0.63

(3.45) are given in Figure 3.29. According to the results in Table 3.4 and Figure 3.20, the performance of CFST with memory is more robust than other fusion algorithms in case of any outlier for such a case.

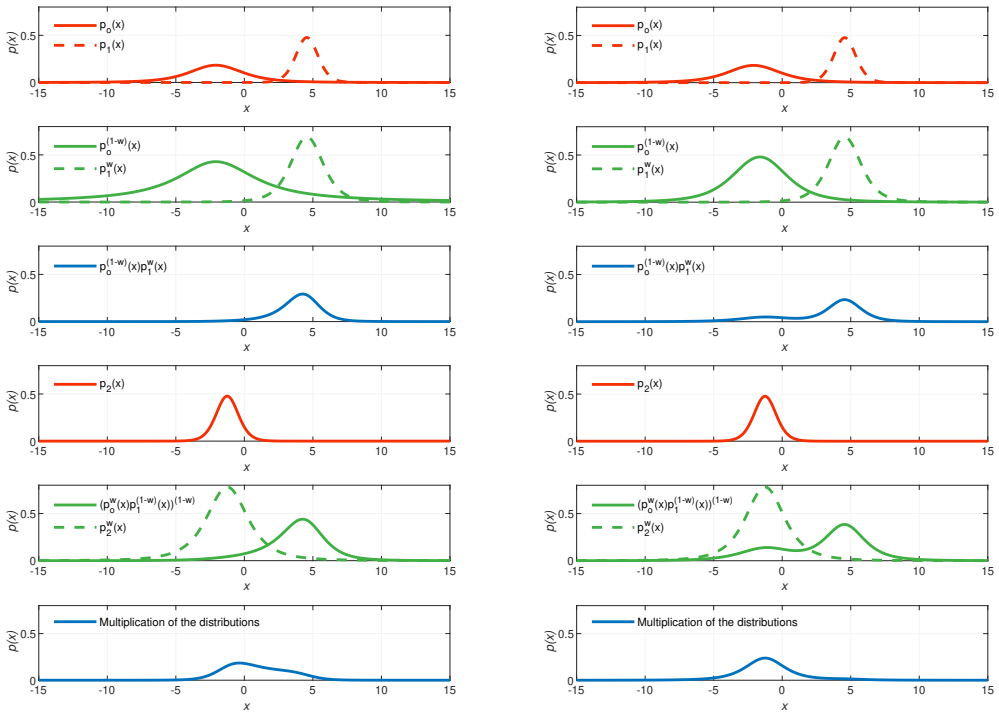
3.3.7.5 One Dimensional Analysis For CFST with memory

In this subsection, the aim is to investigate the steps of CFST algorithm with one dimensional example. The experiments are conducted for CFST and the numerically calculated Chernoff fusion with student t-distribution in a lower dimensional state-space model. The problem is the same as in Section 3.3.6. The figures below illustrate outputs of the three critical steps of CFST with memory as follows

- 1st row of the figures shows the fusion output of the previous time instant after time update, i.e., $p_o(x)$ and the input distributed with student t-distribution, i.e., $p_1(x)$.
- 2nd row of the figures shows the non-integer power of student t-distributions, i.e., $p_o^{(1-w)}(x)$ and $p_1^w(x)$.
- 3rd row of the figures shows the multiplication of the distributions in the second row, i.e., $p_o^{(1-w)}(x)p_1^w(x)$.
- 4th row of the figures shows input distributed with student t-distribution, i.e., $p_2(x)$.

- 5$^{\text{th}}$ row of the figures shows the non-integer power of the student t-mixture in the third row, i.e., $(p_o^{(1-w)}(x)p_1^w(x))^{(1-w)}$ and the non-integer power of the student t-distribution in the forth row, i.e., $p_2^w(x)$

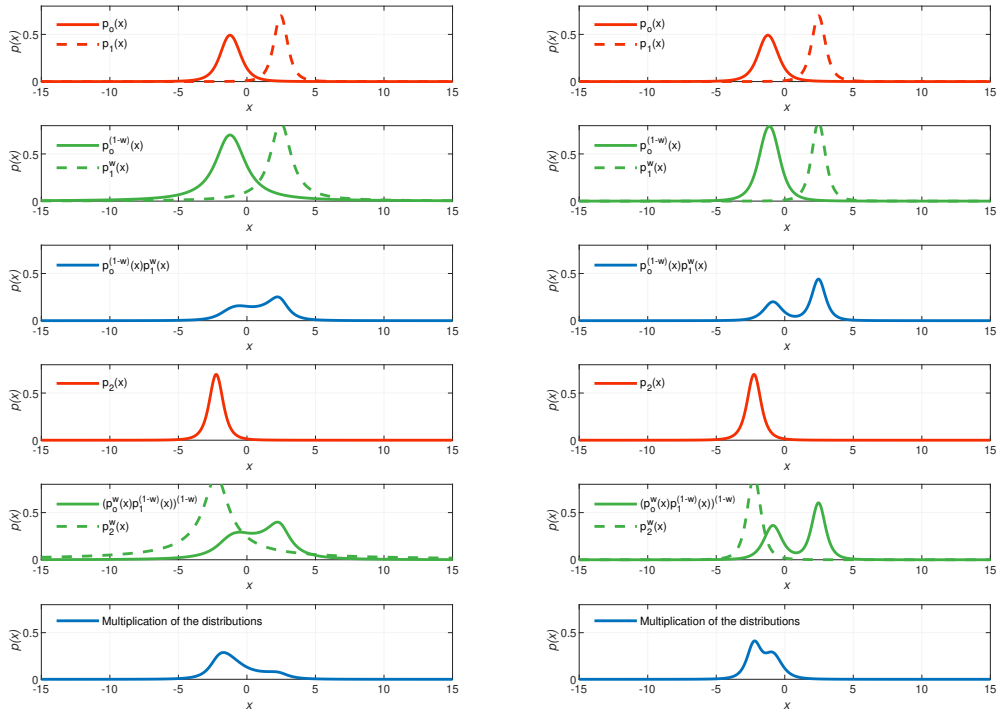
- 6$^{\text{th}}$ row of the figures shows the multiplication of the student t-mixture and student t-distribution in the fifth row, i.e., $(p_o^{(1-w)}(x)p_1^w(x))^{1-w}p_2^w(x)$.



(a) (b)

Figure 3.30: (a) (left figure) shows the numerically calculated Chernoff fusion with student t-distributions outputs. (b) (right figure) shows CFST with memory outputs for each step.

The numerically calculated counterparts of these steps are also available in the figures. The initial degrees of freedom is accepted as $\nu = 10$ for Figure 3.30 and $\nu = 3$ for Figure 3.31. Based on the results in Figure 3.30, the non-integer power of $p_o(x)$ as a result of CFST with memory differs from the one computed numerically since the degrees of freedom of $p_o(x)$ is $\nu = 3$. Since preventing the degrees of freedom from



(a) (b)

Figure 3.31: (a) (left figure) shows the numerically calculated Chernoff fusion with student t-distributions outputs. (b) (right figure) shows CFST with memory outputs for each step.

falling below a certain value (because covariance is available if $\nu > 2$) and the approximations in the non-integer power of the student t-mixture $p_o(x)$, $p_o^{(1-w)}(x)$ of the numerical computation and the CFST with memory are different. Together with these approximations, the approximation in the multiplication of the student t-distributions introduce the difference between the final output. Based on the results in Figure 3.31, the non-integer power of the distributions as a result of CFST and numerical computation are different since preventing the degrees of freedom from falling below a certain value (covariance is available if $\nu > 2$). Furthermore, the situation related to $p_o^{(1-w)}(x)$ in Figure 3.30 is also valid in this case. The approximation in the the non-integer power of the student t-mixture $p_o(x)$ and preventing the degrees of freedom from falling below a certain value create the difference between $p_o^{(1-w)}(x)$ computed

by CFST with memory and numerically. The difference introduced in the non-integer power of student t-distribution and mixture is highlighted more in the final output due to approximation in the multiplication of the distributions.

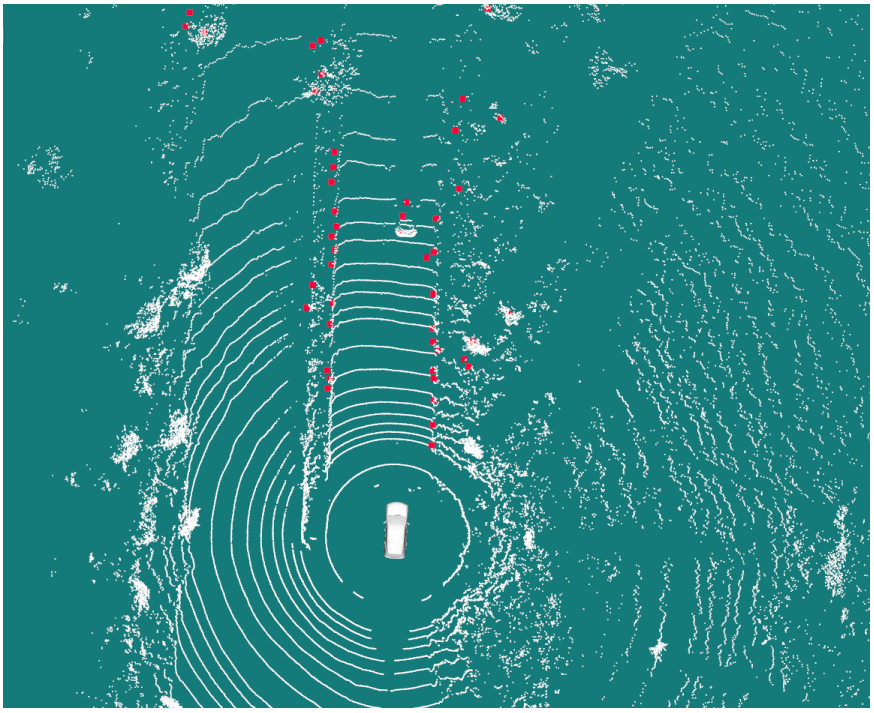
3.4 Real Data Experiments

In this section, the performance of the algorithms is illustrated with real data. The test scenario is carried out in an urban area of Ankara and involves two vehicles, one for data collection and one for testing and validation. The vehicle for data collection is equipped with a camera, LiDAR, and RADAR. A snapshot of the measurements from three different sensors is available in Figure 3.32(a), in which the white and pink points stand for LiDAR and RADAR measurements, respectively. The camera output, which displays the vehicle for testing and validation in Figure 3.32(b), is also accessible.

After collecting raw measurements, each sensor output are processed to extract meaningful information about the vehicle's 2D position used for testing and validation. Due to the dense characteristic of the point cloud, LiDAR is a compelling sensor to solve complex problems such as mapping. Furthermore, the dense point cloud of LiDAR enables us to capture the critical features of the surrounding objects, which results in the capability to use object detection and object classification. As a result, the method for LiDAR point cloud processing is based on an object detection. The center positions of the object detection outcomes are taken as the LiDAR measurement to evaluate the performance of the algorithms.

Unlike LiDAR, the RADAR point cloud is excessively sparse. This characteristic is compensated by using the additional information gathered from the radial speed. The radar measurements originating from the test vehicle are detected by a clustering algorithm. The non-clustered points as a result of the clustering algorithm are eliminated. The mean of the remaining RADAR measurements corresponds to the position measurement of the test vehicle in 2D.

The only sensor that does not provide 2D measurements directly is the camera in the setup. As a result, the position measurements are extracted from camera image by



(a)

(b)

Figure 3.32: (a) displays a snapshot of the raw measurements collected from LiDAR and RADAR, and (b) shows the camera and object detection output.

following two steps called 2D object detection in the camera frame and inverse perspective mapping (IPM). The 2D object detection model trained for detecting vehicle and person classes is used to extract bounding boxes. The working principle of the

IPM is strictly related to the camera model, which is described by the intrinsic and extrinsic parameters. Extrinsic parameters define the relative position and orientation of the camera with respect to the world frame. On the other hand, intrinsic parameters, which describe how the world warped into the camera image, are used to construct a transformation matrix from the world frame to the image frame. Applying transformation to the road creates a map between the image frame and the road, which enables us to generate 2D position measurements from camera pixels. The bottom midpoint of the bounding boxes is given to IPM as input, and the result is the 2D position measurement of the object of interest.

After processing raw measurements of each sensor, an important step is to transform measurements from the sensor frame to the world frame. The configuration of the sensors regarding the world frame is a critical concept. Any oversight between a sensor's actual and known pose respecting the world frame creates ambiguity if the other sensors' pose is determined precisely enough. In order to evaluate the performance of the algorithms, four sets of measurements are collected, two representing the ideal case and the others having the ill-formed configuration, which creates ambiguity due to an intentional discrepancy between the actual and known orientation of the camera. Although outputs of IPM properly reflect the test vehicle's position in the camera frame, the wrong configuration prevents a proper connection between the processing phase and the algorithms like the Kalman filter and SF.

Before proceeding with the algorithms, another essential point is synchronizing the measurements from different sensors because the camera, RADAR, and LiDAR run at 30 Hz, 20 Hz, and 10 Hz, respectively. Furthermore, a sensor known as vbox that is installed on the test vehicle to obtain the actual position and velocity runs at 10 Hz. By computing the sensor readings at vbox timestamps via linear interpolation, the sensor measurements are synchronized with the vbox for performance evaluation.

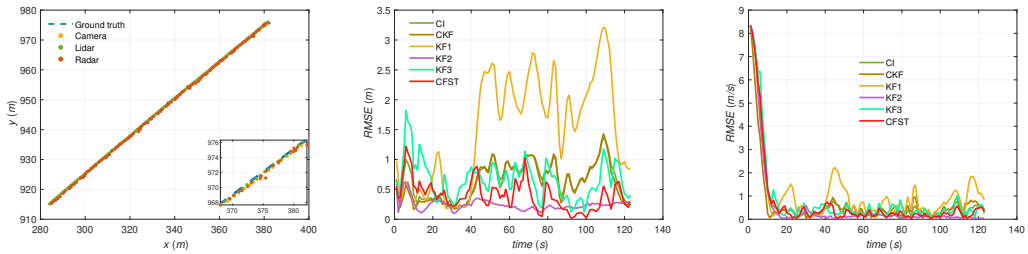
The adjusted and synchronized sensor measurements are sent to the Kalman filter for CI and CFST and transferred directly to SF. The state of the test vehicle $x_k = [p_{x,k} \quad p_{y,k} \quad v_{x,k} \quad v_{y,k}]^{\mathrm{T}}$ is defined with the planar position and velocity as in the simulations. The state-space model is the same as the one described in (3.44) with the process noise variances $\sigma_x^2 = 25$ and $\sigma_y^2 = 9$, and the measurement noise

variances $\sigma_x^2 = \sigma_y^2 = 1$.

The outcomes of the multi-sensor fusion algorithms are illustrated with two distinct configurations; one is the correct configuration, while the other is erroneous.

3.4.1 Case 1

In this case, all the sensors' known translations and rotations concerning the world frame are known precisely enough to collect meaningful measurements from the surroundings. As a result, there is no ambiguity due to measurements, which can be labeled as a convenient case for fusion. Figure 3.33(a) illustrate the sensor measurements and the ground truth of the target trajectory.



(a) (b) (c)

Figure 3.33: (a) shows the sensor measurements and ground truth. (b) displays the position RMSE of the algorithms over time. (c) shows the velocity RMSE of the algorithms over time.

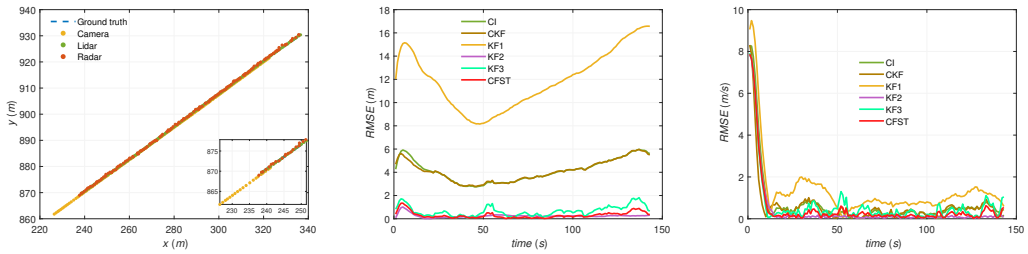
Table 3.9: Average RMSE values of the position and velocity for the Case 1.

	RMSE (*m*)	**RMSE** (*m/s*)
CI	0.72	1.68
CKF	0.69	1.44
KF1	1.75	1.84
KF2	0.26	1.59
KF3	1.73	1.77
CFST	0.47	1.67

The average RMSE values of the scenario in Case 1 are given in Table 3.9. The position and velocity RMSE values calculated using (3.45) are given in Figure 3.33(b) and Figure 3.33(c). According to the results in Table 3.9 and Figure 3.33, the performance of the fusion algorithms are comparable. On the other hand, one of the local Kalman filters outperforms the fusion methods.

3.4.2 Case 2

Although the pose and orientation of LiDAR and RADAR concerning the world frame are known precisely enough in this scenario, there is a mismatch between the camera's actual and known orientation regarding the world frame. Such a disparity may cause massive positional mistakes in IPM output, resulting in ambiguity that the approaches must resolve. Figure 3.34(a) illustrate the sensor measurements and the ground truth of the target trajectory.



(a) (b) (c)

Figure 3.34: (a) shows the sensor measurements and ground truth. (b) displays the position RMSE of the algorithms over time. (c) shows the velocity RMSE of the algorithms over time.

The average RMSE values of the scenario in Case 2 are given in Table 3.10. The position and velocity RMSE values calculated using (3.45) are given in Figure 3.34(b) and Figure 3.34(c). According to the results in Table 3.10 and Figure 3.34, CFST performs better than other fusion algorithms in case of ambiguity. Further, although one of the local Kalman filters outperforms the fusion methods, the performance of CFST is comparable with that local Kalman filter performance.

Table 3.10: Average RMSE values of the position and velocity for the Case 2.

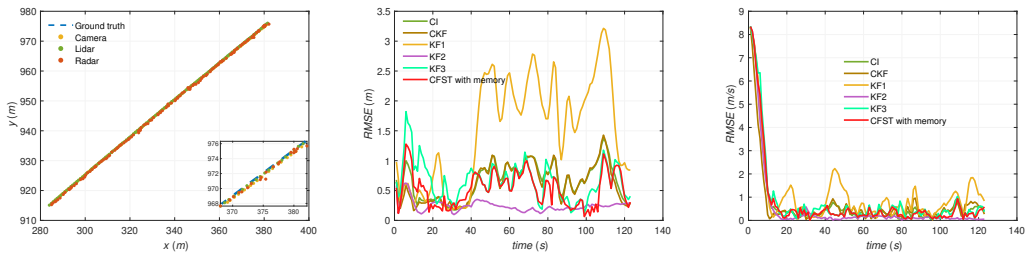
	RMSE (*m*)	**RMSE** (*m/s*)
CI	4.26	1.54
CKF	4.22	1.30
KF1	12.22	2.07
KF2	0.29	1.35
KF3	0.78	1.40
CFST	0.41	1.35

3.4.3 Discussion: CFST with memory

In this part, the real data experiments are also conducted for the performance evaluation of CFST with memory.

3.4.3.1 Case 1

In this situation, all of the sensors' known translations and rotations with respect to the world frame are exact enough to collect useful measurements from the environment. Figure 3.35(a) illustrate the sensor measurements and the ground truth of the target trajectory.



(a) (b) (c)

Figure 3.35: (a) shows the sensor measurements and ground truth. (b) displays the position RMSE of the algorithms over time. (c) shows the velocity RMSE of the algorithms over time.

Table 3.11: Average RMSE values of the position and velocity for the Case 1.

	RMSE (*m*)	**RMSE** (*m/s*)
CI	0.72	1.68
CKF	0.69	1.44
KF1	1.75	1.84
KF2	0.26	1.59
KF3	1.73	1.77
CFST with memory	0.59	1.69

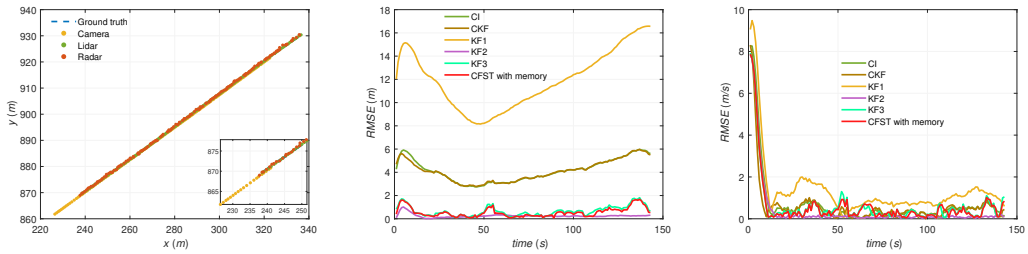
The average RMSE values of the scenario in Case 1 are given in Table 3.11. The position and velocity RMSE values calculated using (3.45) are given in Figure 3.35(b) and Figure 3.35(c). According to the results in Table 3.11 and Figure 3.35, although the performance of the fusion algorithms are comparable, one of the local Kalman filters outperforms the fusion methods.

3.4.3.2 Case 2

Although the pose and orientation of LiDAR and RADAR concerning the world frame are known precisely enough in this scenario, there is a mismatch between the camera's actual and known orientation regarding the world frame. Such a disparity may cause massive positional mistakes in IPM output, resulting in ambiguity that the approaches must resolve.

Table 3.12: Average RMSE values of the position and velocity for the Case 2.

	RMSE (*m*)	**RMSE** (*m/s*)
CI	4.26	1.54
CKF	4.22	1.30
KF1	12.22	2.07
KF2	0.29	1.35
KF3	0.78	1.40
CFST	0.67	1.39



(a) (b) (c)

Figure 3.36: (a) shows the sensor measurements and ground truth. (b) displays the position RMSE of the algorithms over time. (c) shows the velocity RMSE of the algorithms over time.

The average RMSE values of the scenario in Case 2 are given in Table 3.12. The position and velocity RMSE values calculated using (3.45) are given in Figure 3.36(b) and Figure 3.36(c). According to the results in Table 3.12 and Figure 3.36, CFST outperforms other fusion algorithms, but one of the local Kalman filters outperforms the fusion methods.

3.5 Conclusion

This chapter investigates the collaboration of the Chernoff fusion with the student t-distribution (CFST). The simulations and real data experiments are conducted to evaluate the performance of CFST. The various algorithms, such as covariance intersection (CI), centralized Kalman filter (CKF), and local Kalman filters, are also included in the performance evaluation for comparison. According to the simulation results, CI and CKF outperform CFST if the methods precisely know the sensor characteristics. On the other hand, in case of a mismatch between the actual and known characteristics of one sensor, CFST performs better than CI, CKF, and local Kalman filters for conflict resolution. According to the real data experiments, the performance of the fusion methods are comparable if the sensor characteristics are known by the methods precisely enough. Moreover, the results of the real data experiments are similar to the simulations in case of conflict between the known and actual sensor characteristics. Additionally, there is a CFST substitute in the form of additional input. The

method is known as CFST with memory if the fusion output of the preceding instant after time update is sent to the CFST algorithm. According to the simulations, CI and CKF perform slightly better than CFST with memory in the case of the known sensor characteristics. Meanwhile, CFST with memory outperforms CI and CKF when there is a mismatch between the actual and known sensor characteristics. Based on the real data experiments, the performance of CI, CKF, and CFST with memory are similar when there is no conflict. In case of sensor failure, CFST with memory outperforms CI and CKF.

CHAPTER 4

VARIATIONAL MEASUREMENT UPDATE FOR EXTENDED OBJECT TRACKING USING GAUSSIAN PROCESSES

Conventional tracking methods assume that sensors are capable of detecting objects as point sources without extent. As a result, the availability of at most one measurement per scan is a common assumption. Instead of these assumptions, when a sensor is adequate to get multiple measurements to detect an object, different approaches for tracking in terms of kinematics and extent are introduced.

The approach described in [48, 49] is based on Bayesian recursion and involves time update and measurement update with the joint estimation of kinematics and extent. It models the target extent using positive definite matrices distributed with inverse Wishart distribution. A random matrix-based model [50] approximates densities using minimization of Kullback-Leibler divergence for extent and differs from [48, 49] on the prediction update method. Further, this method allows for taking into account a target's potential rotation. Another random matrix-based model is introduced in [51], describing possible dynamics and extent adjustments using different models. [52, 53] combine extended target tracking with spatial distribution models. The proposed approach specifies measurements as samples from the probability distribution related to the target extent. [54, 55] incorporate set-theoretic and spatial distribution models to produce a random hypersurface model. The proposed approach can estimate an arbitrary shape of the target in addition to tracking the kinematics. [56, 57] assign specific geometric shapes to the extent of the target. [56] employs predicted measurements and corresponding innovation covariances. The proposed approach computes the appropriate likelihood functions under the assumption that feasible target types include a rectangle and an ellipse. On the other hand, [57] offers various models con-

structed by two separate motion models, as well as two different measurement models based on points and lines. [58] makes use of the variational inference approximation to achieve the measurement update for the extended targets with random matrices. [59] models the target extent with multiple ellipses and solves the inference problem considering the sequential Monte Carlo framework.

This section focuses on an alternative measurement update constructed with the variational inference for the Gaussian process-based extended target tracking models. The performance of the alternative measurement update is evaluated by comparison with [4] through simulations and real data experiments.

4.1 Problem Definition

The fundamental objective is to track the position and extent of a target using noisy measurements. The two-dimensional position measurements $\{z_{k,1} \cdots z_{k,n_k}\}$ are generated from the target contour with additive white noise at each time step k. These finite numbers of measurements are identically distributed and independent from each other. We can derive details about the object's contour thanks to the availability of multiple measurements. At this point, describing the target extent is achieved using star-convex shapes as proposed in [54]. This definition facilitates contour representation with a radial function corresponding to a basis angle such as $r = f(\theta)$. Based on this model, the basis angles and corresponding radial distances describe the target extent.

In early work, [4], the link between the measurements and the target extent is formed using the Gaussian process, which models the contour as

$$f(\theta) \sim \mathit{GP}(r, k(\theta, \bar{\theta})), \quad r \sim \mathcal{N}(0, \sigma_r^2), \tag{4.1}$$

where mean function of Gaussian process, r, is the estimate of the radial function. Covariance function $k(\theta, \bar{\theta})$ determines the correlation level between basis angles θ and $\bar{\theta}$. Due to the periodic characteristic of the target extent defined via star-convex shape, the kernel function defined in 2.59 is utilized to make $f(\theta)$ and $f(\theta + 2\pi)$ perfectly correlated. In addition, zero-valued mean function performs well for Gaussian

process, which results in the following reformulation

$$f(\theta) \sim \mathit{GP}(0, k(\theta, \bar{\theta}) + \sigma_r^2). \tag{4.2}$$

After constructing the Gaussian process model to represent the extent, derivation of a state space model is another crucial step in applying Kalman filter-based approaches to that state space model as in [4].

4.1.1 State Space Model

In this subsection, the complete state space model including process model and measurement model is introduced. Initially, state vector for this model can be written as

$$x_k = \begin{bmatrix} (\bar{x}_k)^\top & (f_k)^\top \end{bmatrix}^\top, \tag{4.3a}$$

where $\bar{x}_k$ and f_k stand for kinematics and extent of the target. $\bar{x}_k$ can be expressed in more detail as given

$$\bar{x}_k = \begin{bmatrix} (x_k^c)^\top & \psi_k & (x_k^v)^\top \end{bmatrix}^\top, \tag{4.3b}$$

where x_k^c, ψ_k and x_k^v represent center position, orientation and state elements related to velocity and angular velocity, respectively.

4.1.1.1 Process Model

The following describes the process model to explain a version of the position and extent of the target based on various assumptions at each stage

$$x_{k+1} = F x_k + w_k, \;\; w_k \sim \mathcal{N}(0,\, Q_k) \tag{4.4a}$$

where

$$F = \begin{bmatrix} F_k & 0 \\ 0 & F_f \end{bmatrix}, \;\; Q = \begin{bmatrix} Q_k & 0 \\ 0 & Q_f \end{bmatrix}. \tag{4.4b}$$

The constant velocity model is used to simulate the target's dynamic behavior. Applying the constant velocity model requires the following inputs

$$F_k = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \otimes I_3, \;\; Q_k = \begin{bmatrix} \frac{\Delta T^3}{3} & \frac{\Delta T^2}{2} \\ \frac{\Delta T^2}{2} & \Delta T \end{bmatrix} \otimes \begin{bmatrix} \sigma_c^2 & 0 & 0 \\ 0 & \sigma_c^2 & 0 \\ 0 & 0 & \sigma_\psi^2 \end{bmatrix}, \tag{4.5}$$

where σ_c and σ_ψ are standard deviations for center position and orientation, respectively. Moreover, describing dynamics of extent can be accomplished using the following

$$F_f = I, \;\; Q_f = \left(\alpha^{-1} - 1\right) P^f_{k|k}, \tag{4.6}$$

where α is a factor that scales up the estimated covariance of the extent $P^f_{k|k}$ to compute the predicted covariance of the extent $P^f_{k+1|k}$.

4.1.1.2 Measurement Model

Depending on the way of describing the target contour with a star-convex shape, the measurement model to explain the generation of noisy measurements from the target contour can be expanded as

$$z_{k,l} = x^c_k + p(\theta_{k,l}) f(\theta_{k,l}) + e_{k,l}, \tag{4.7}$$

where x^c_k is the center of target at time index *k*; $\{z_{k,l}\}_{l=1}^{n_k}$ are the n_k measurements collected at time index k; $\{\theta_{k,l}\}_{l=1}^{n_k}$ represent the angles expressing the source of the measurements on the target contour; $e_{k,l} \sim \mathcal{N}(0, R)$ denotes the zero mean Gaussian measurement noise with covariance *R*; and $p(\theta_{k,l})$, orientation vector, is defined as

$$p(\theta_{k,l}) = \begin{bmatrix} \cos(\theta_{k,l}) \\ \sin(\theta_{k,l}) \end{bmatrix}. \tag{4.8}$$

The association between the kinematics and the extent related parameters such as measurements is defined in two different coordinate frames. The first is the global coordinate frame, which includes targets and sensor measurements, and the second is the local coordinate frame, which is based on how the target is oriented with respect to the global frame. The relation between measurements $\{z_{k,l}\}_{l=1}^{n_k}$ and the center of target x^c_k is defined in both the global and local target coordinate frames as follows

$$\theta^G_{k,l}(x^c_k) = \angle(z_{k,l} - x^c_k), \tag{4.9a}$$

$$\theta^L_{k,l}(x^c_k, \psi_k) = \theta^G_{k,l}(x^c_k) - \psi_k, \tag{4.9b}$$

where ψ_k stands for orientation of the target. In the light of these angle definitions, the measurement model can be written as

$$z_{k,l} = x^c_k + p_{k,l}(x^c_k) f(\theta^L_{k,l}(x^c_k, \psi_k)) + \bar{e}_{k,l}, \tag{4.10}$$

where the extent model $f(\theta^L_{k,l}(x^c_k, \psi_k))$ established with respect to the local frame is transferred to the global frame by multiplying with $p_{k,l}(x^c_k)$ to arrange rotation and then by adding x^c_k to set translation. Radial function $f(\theta^L_{k,l}(x^c_k, \psi_k))$ used to explain the target extent can be expanded using the Gaussian Process description in 2.65, 2.66 as follows

$$f(\theta^L_{k,l}(x^c_k, \psi_k)) = H^f\left(\theta^L_{k,l}(x^c_k, \psi_k)\right) x^f_k + e^f_{k,l}, \tag{4.11}$$

Furthermore, when these angle definitions are used to clarify measurement model, the orientation vector $p(\theta_{k,l})$ can be reformulated for simplicity

$$p_{k,l}(x^c_k) = p(\theta^G_{k,l}(x^c_k)) = \frac{z_{k,l} - x^c_k}{\|z_{k,l} - x^c_k\|}. \tag{4.12}$$

These expansions enable us to write measurement equation as in 4.13,

$$\begin{aligned}
z_{k,l} &= x^c_k + p_{k,l}(x^c_k)\left[H^f\left(\theta^L_{k,l}(x^c_k, \psi_k)\right) x^f_k + e^f_{k,l}\right] + \bar{e}_{k,l}, \\
&= h_{k,l}(x_k) + e_{k,l}, && \text{(4.13a)} \\
h_{k,l}(x_k) &= x^c_k + \tilde{H}^f_l(\bar{x}_k) x^f_k, && \text{(4.13b)} \\
\tilde{H}^f_l(\bar{x}_k) &= p_{k,l}(x^c_k) H^f\left(\theta^L_{k,l}(x^c_k, \psi_k)\right), && \text{(4.13c)} \\
e_{k,l} &= p_{k,l}(x^c_k) e^f_{k,l} + \bar{e}_{k,l}, \quad e_{k,l} \sim \mathcal{N}(0,\ R_{k,l}), && \text{(4.13d)} \\
R_{k,l} &= p_{k,l}(x^c_k) R^f\left(\theta^L_{k,l}\left(x^c_k, \psi_k\right)\right) p_{k,l}(x^c_k)^T. && \text{(4.13e)}
\end{aligned}$$

The model described in 4.13 is expanded with a model in which all the measurements in one scan is utilized together to update estimate. The corresponding augmented version of the measurement model can be written as

$$z_k = h_k(x_k) + e_k, \quad e_k \sim \mathcal{N}(0,\ R_k), \tag{4.14}$$

where

$$\begin{aligned}
z_k &= \left[z^\top_{k,1}, \ \cdots, \ z^\top_{k,n_k}\right]^\top, && \text{(4.15a)} \\
h_k(x_k) &= \left[h_{k,1}(x_k)^\top, \ \cdots, \ h_{k,n_k}(x_k)^\top\right]^\top, && \text{(4.15b)} \\
R_k &= \mathrm{diag}[R_{k,1}, \cdots, R_{k,n_k}]. && \text{(4.15c)}
\end{aligned}$$

4.2 Measurement Update Based on Variational Inference

The variational inference approaches problem as solving an optimization problem. We aim to find an approximate solution for the posterior $p(\bar{x}_k, f_k|z_{1:k})$ by starting with the following approximation

$$p(\bar{x}_k, f_k|z_{1:k}) \approx q(\bar{x}_k, f_k) = q_x(\bar{x}_k)q_f(f_k), \tag{4.16}$$

where the $q_x(\bar{x}_k)$ and $q_f(f_k)$ stand for approximate densities for kinematics and extent, respectively. The concept is founded on minimizing the disparity between the approximate and actual posteriors. The distance measure is defined with the Kullback-Leibler divergence, i.e., $\text{KL}(p(x)||q((x)) = \int q\text{log}(\frac{q}{p})dx$, resulting in the following optimization problem statement

$$q_x(\bar{x}_k), q_f(f_k) = \text{arg min KL}[q_x(\bar{x}_k)q_f(f_k)||p(\bar{x}_k, x_k^f|z_{1:k})]. \tag{4.17}$$

The solution to (4.17) is based on an iterative procedure. The key aspect is to update one of the factorized approximate densities while keeping the other one fixed on its last version. Analytical solutions for the approximate densities are given as

$$\text{log } q_x(\bar{x}_k) = E_{q_f}[\text{log}p(z_k, \bar{x}_k, f_k|z_{1:k-1})] + c_1, \tag{4.18a}$$

$$\text{log } q_f(f_k) = E_{q_x}[\text{log}p(z_k, \bar{x}_k, f_k|z_{1:k-1})] + c_2. \tag{4.18b}$$

The joint density $p(z_k, \bar{x}_k, f_k|z_{1:k-1})$ to compute approximate solutions as mentioned in (4.18) is factorized as follows

$$p(z_k, \bar{x}_k, f_k|z_{1:k-1}) = p(z_k|\bar{x}_k, f_k)p(\bar{x}_k, f_k|z_{1:k-1}), \tag{4.19}$$

where $p(z_k|\bar{x}_k, f_k)$ and $p(\bar{x}_k, f_k|z_{1:k-1})$ stand for the measurement likelihood and the predicted state density, respectively. The predicted state density can be factorized as follows

$$p(\bar{x}_k, f_k|z_{1:k-1}) = p(\bar{x}_k|z_{1:k-1})p(f_k|z_{1:k-1}) \tag{4.20}$$

, where

$$p(\bar{x}_k|z_{1:k-1}) \sim \mathcal{N}(\bar{x}_k; \mu^x_{k|k-1}, P^x_{k|k-1}), \tag{4.21a}$$

$$p(f_k|z_{1:k-1}) \sim \mathcal{N}(f_k; \mu^f_{k|k-1}, P^f_{k|k-1}). \tag{4.21b}$$

Based on this model, approximate posteriors are also Gaussian since posterior and prior should be in the same form for the recursion. As a result, approximate posterior target density can be modeled as

$$q_x^{i+1}(\bar{x}_k) \sim \mathcal{N}(\bar{x}_k; \mu_x^{i+1}, P_x^{i+1}), \tag{4.22a}$$

$$q_f^{i+1}(f_k) \sim \mathcal{N}(f_k; \mu_f^{i+1}, P_f^{i+1}), \tag{4.22b}$$

where $q_x^{i+1}(\bar{x}_k)$ and $q_x^{i+1}(f_k)$ correspond to the estimated densities at the $(i+1)^{th}$ iteration for the kinematics and the extent. The nonlinear characteristics of the measurement model defined in 4.14 violate the condition that the posterior is in the same form as the prior to have a tractable algorithm finally. The selected method to resolve this issue depends on the computation of which approximate density. The measurement model is approximated by its first-order Taylor series expansion for computing approximate kinematic density $q_x^{i+1}(\bar{x}_k)$. On the other hand, the unscented transform is used to deal with the nonlinear measurement model to calculate approximate extent density $q_f^{i+1}(f_k)$. The details related to the derivation of the analytical expression for the approximate kinematics and extent densities are provided in [60]. The measurement update equations for the kinematics' mean and the covariance at the $(i)^{th}$ iteration are as follows

$$\mu_x^{i+1} = \mu_{k|k-1}^x + K_k^x(z_k - g^i - A_x^{i+1}\mu_{k|k-1}^x - B_x^{i+1}\mu_f^{i+1}), \tag{4.23a}$$

$$P_x^{i+1} = P_{k|k-1}^x - K_k^x A_x^{i+1} P_{k|k-1}^x, \tag{4.23b}$$

where

$$g^i = h_k(\mu_x^i, \mu_f^i) - A_x^i \mu_x^i - B_x^i \mu_f^i, \tag{4.24a}$$

$$A_x^{i+1} = \frac{\partial h}{\partial \bar{x}_k}|_{\bar{x}_k=\mu_x^i, f_k=\mu_f^i}, \tag{4.24b}$$

$$B_x^{i+1} = \frac{\partial h}{\partial f_k}|_{\bar{x}_k=\mu_x^i, f_k=\mu_f^i}, \tag{4.24c}$$

$$S_k^x = A_x^{i+1} P_{k|k-1}^x (A_x^{i+1})^\top + R_k, \tag{4.24d}$$

$$K_k^x = P_{k|k-1}^x (A_x^{i+1})^\top (S_k^x)^{-1}. \tag{4.24e}$$

The measurement update of the extent mean and covariance at the $(i)^{th}$ iteration is achieved as

$$\mu_f^{i+1} = \left(\Delta + (P_{k|k-1}^f)^{-1}\right)^{-1} \left(\delta + (P_{k|k-1}^f)^{-1}\mu_{k|k-1}^f\right), \tag{4.25a}$$

$$P_f^{i+1} = \left(\Delta + (P_{k|k-1}^f)^{-1}\right)^{-1}, \tag{4.25b}$$

where

$$\Delta \approx \sum_{j=0}^{M} \pi^j g_1(\bar{x}_k^j),\ \delta \approx \sum_{j=0}^{M} \pi^j g_2(\bar{x}_k^j), \tag{4.26}$$

$$g_1(\bar{x}_k) = \tilde{H}_l^f(\bar{x}_k)^\top R_k^{-1} \tilde{H}_l^f(\bar{x}_k), \tag{4.27}$$

$$g_2(\bar{x}_k) = \tilde{H}_l^f(\bar{x}_k)^\top R_k^{-1}(z_k - x_k^c), \tag{4.28}$$

where $\{\bar{\mathbf{x}}_k^j\}_{j=0}^M$ and $\{\pi^j\}_{j=0}^M$ stand for the sigma-points and the corresponding weights, respectively.

4.3 Simulation Results

In this section, the performance of the inference method based on variational inference is evaluated and compared with the method presented in [4] through simulations. Three separate experiments are conducted to present various challenges for a tracking application. In the experiments, a target of various shapes travels along a path that combines linear and curved motions. The first experiment represents the ideal case since measurements are collected with a sufficiently small sampling time, and the dynamic model can describe the target's motion. Further, the level of uncertainty about the target's motion is as tiny as possible. In the second experiment, the uncertainty about the target's motion is much more apparent than in the first experiment, and the dynamic model is insufficient to describe the target's motion. The last experiment is the same as the first one except for having a more considerable sampling time. Different simulations are performed 100 times to assess the estimation results of the methods. The algorithms are evaluated using different realizations of measurements in the Monte Carlo runs. The evaluation of estimation results related to extent is based on a measure called Intersection-Over-Union (IOU). If the area of the estimated target and the true target are represented by $\hat{A}$ and A_0, respectively, IOU is determined by computing the intersection and union of these two areas and calculating ratio of intersection and union as follows

$$\text{IOU}(\hat{A}, A_0) = \frac{\text{area}(\hat{A} \cap A_0)}{\text{area}(\hat{A} \cup A_0)}. \tag{4.29}$$

$\text{IOU}(\hat{\text{A}}, \text{A}_0) \in [0, 1]$, where 1 means perfect overlap between estimated and true object and 0 means no overlap in any case. The evaluation of estimation related to

target position is achieved using root-mean-square-error (RMSE) defined in (3.45) and (3.46).

Throughout the simulations, the measurements are originated from the contour of the moving target whose shape can be rectangular (T1), triangular (T2) and plus (T3). Further, the number of measurements are Poisson distributed with mean 15, and the measurement noise standard deviation is set to $\sigma_v = 0.2$. The motion and the GP parameters used in the experiments are provided in Table 4.1. The ideal case, also labeled E1, is accepted as a base regarding the motion and GP parameters. In the case with the model mismatch labeled as E2, the same motion and GP parameters are used with the ideal case except for the target's speed. Similarly, the sampling time is the only parameter difference between the case with the more significant sampling time labeled as E3 and the ideal case.

Table 4.1: Motion and GP parameters used in the experiments. P1 represents the motion parameters, and P2 stands for the GP parameters.

	P1					**P2**		
	ΔT	σ_c	σ_ψ	α	***v* (m/s)**	σ_r	σ_f	***l***
E1	1	0.1	0.01	0.99	0.5	0.6	3	$\frac{\pi}{4}$
E2	1	0.1	0.01	0.99	2	0.6	3	$\frac{\pi}{4}$
E3	4	0.1	0.01	0.99	0.5	0.6	3	$\frac{\pi}{4}$

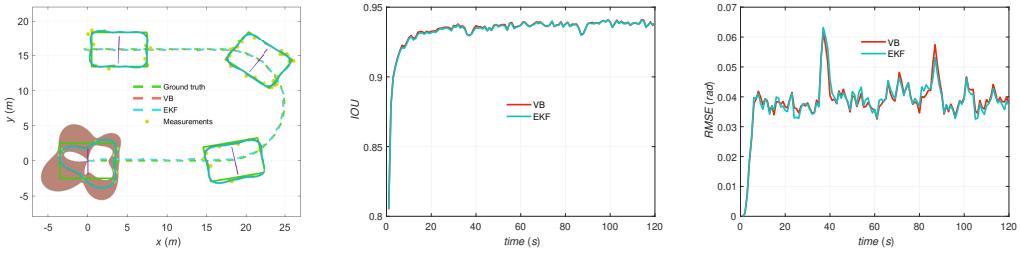
4.3.1 The ideal case

The ideal case corresponds to the filtering with a sufficiently small sampling time and as little uncertainty as possible about the target's motion characteristics. Figure 4.1 illustrates the performance of the algorithms for the target whose shape is rectangle. More specifically, Figure 4.1(a) illustrates the realization of the extended target tracking results obtained by the measurement update based on variational inference (VB) and extended Kalman filter (EKF). Furthermore, Figure 4.1(b) and Figure 4.1(c) display the IOU and the RMSE for the orientation angle of the algorithms over 100 Monte Carlo runs for each instant, respectively. The counterparts of the Figure 4.1 for the targets whose shapes are triangle and plus are given in Figure 4.2 and Fig-

ure 4.3, respectively. The average IOU and RMSE values over 100 Monte Carlo runs are given in Table 4.2 for each target with various shapes.

Table 4.2: Average IOU and RMSE values of the orientation angle over 100 Monte Carlo runs for the ideal case. T1, T2 and T3 stand for rectangle, triangle and plus, respectively.

	T1		**T2**		**T3**	
	IOU	**RMSE** (*rad*)	**IOU**	**RMSE** (*rad*)	**IOU**	**RMSE** (*rad*)
EKF	0.933	0.038	0.878	0.037	0.907	0.027
VB	0.934	0.038	0.867	0.037	0.908	0.027



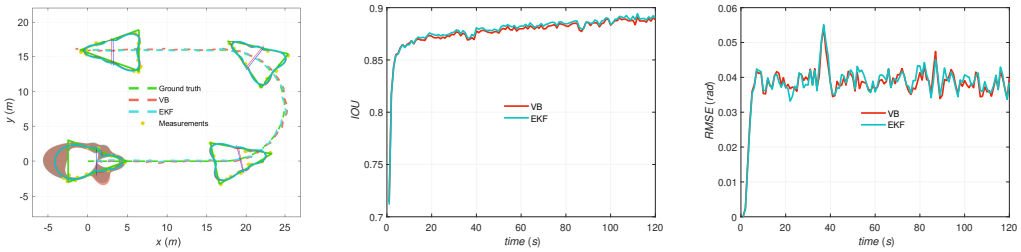
(a) (b) (c)

Figure 4.1: (a) shows a typical realization of the simulation for the ideal case with rectangle-shaped object. (b) and (c) display IOU and RMSE for orientation angles over 100 Monte Carlo runs at each instant, respectively.

Based on the results of Figures 4.1,4.2 and 4.3 and also Table 4.2, the performance of the both methods are almost the same and successful in terms of the IOU and RMSE of the orientation angle.

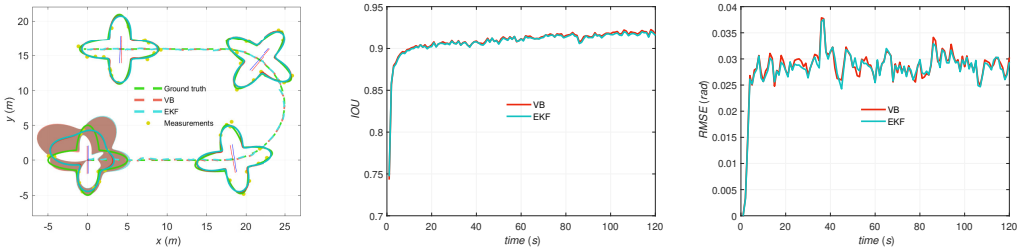
4.3.2 The case with the model mismatch

The mismatch between the actual motion characteristics of the target and the dynamic model assumption is a common condition in target tracking scenarios. The response of the algorithms in such a case is examined by setting a higher target speed that the dynamic model does not justify. The performance of the algorithms for the rectangle-



(a) (b) (c)

Figure 4.2: (a) displays a typical realization of the simulation for the ideal case with triangle-shaped object. (b) and (c) show IOU and RMSE for orientation angles over 100 Monte Carlo runs at each instant, respectively.



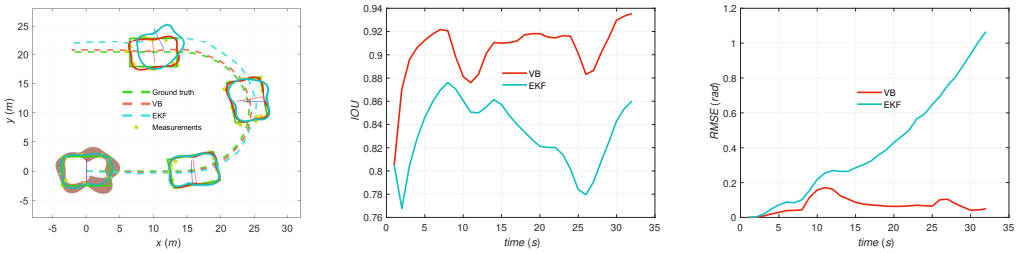
(a) (b) (c)

Figure 4.3: (a) displays a typical realization of the simulation for the ideal case with plus-shaped object. (b) and (c) show IOU and RMSE for orientation angles over 100 Monte Carlo runs at each instant, respectively.

shaped target is shown in Figure 4.4. In more detail, Figure 4.4(a) shows realizations of the VB and EKF based algorithms. The IOU and the RMSE for the orientation angle of the algorithms over 100 Monte Carlo runs for each instant are provided in Figure 4.4(b) and Figure 4.4(c), respectively. Figure 4.5 and Figure 4.6 show the performance of the algorithms for the targets whose shapes are triangle and plus. Table 4.3 presents the average IOU and RMSE values over 100 Monte Carlo runs for each target with the specified shapes.

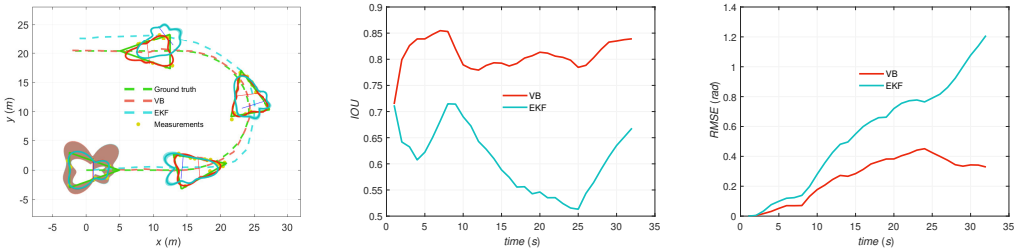
Table 4.3: Average IOU and RMSE values of the orientation angle over 100 Monte Carlo runs for the case with the model mismatch. T1, T2 and T3 stand for rectangle, triangle and plus, respectively.

	T1		**T2**		**T3**	
	IOU	**RMSE** (*rad*)	**IOU**	**RMSE** (*rad*)	**IOU**	**RMSE** (*rad*)
EKF	0.83	0.27	0.61	0.49	0.54	0.88
VB	0.90	0.08	0.81	0.15	0.86	0.06



(a) (b) (c)

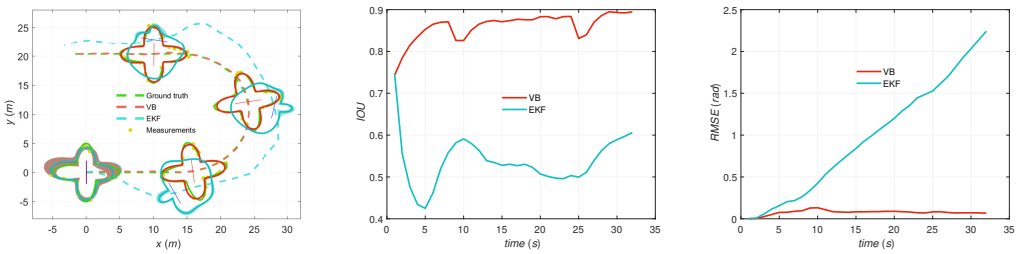
Figure 4.4: (a) displays a typical realization of the simulation for the model mismatch case with rectangle-shaped object. (b) and (c) show IOU and RMSE for orientation angles over 100 Monte Carlo runs at each instant, respectively.



(a) (b) (c)

Figure 4.5: (a) displays a typical realization of the simulation for the model mismatch case with triangle-shaped object. (b) and (c) show IOU and RMSE for orientation angles over 100 Monte Carlo runs at each instant, respectively.

The measurement update framework based on variational inference outperforms the one with the extended Kalman filter in estimating the extent and the orientation angle,



(a) (b) (c)

Figure 4.6: (a) displays a typical realization of the simulation for the model mismatch case with plus-shaped object. (b) and (c) show IOU and RMSE for orientation angles over 100 Monte Carlo runs at each instant, respectively.

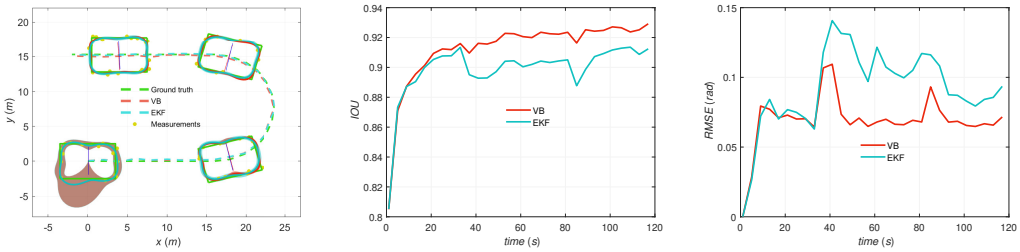
as shown by the findings of Figures 4.4, 4.5, and 4.6, and also Table 4.3.

4.3.3 The case with the larger sampling time

In this case, the performance of the algorithms is investigated by increasing the sampling time with respect to the ideal case. The results for the rectangle-shaped target is presented in Figure 4.7. Specifically, Figure 4.7(a) shows realization of the VB and EKF based algorithms. The IOU and the RMSE for the orientation angle of the algorithms over 100 Monte Carlo runs for each instant are provided in Figure 4.7(b) and Figure 4.7(c), respectively. Figure 4.8 and Figure 4.9 display the performance of the algorithms for the triangle-shaped and plus-shaped targets. Table 4.4 demonstrates the average IOU and RMSE values over 100 Monte Carlo runs for each target with the specified shapes.

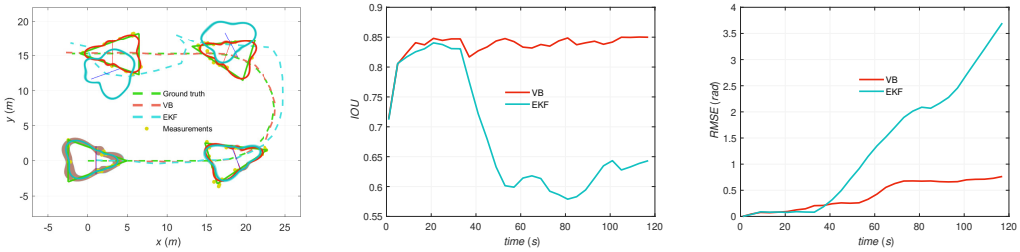
Table 4.4: Average IOU and RMSE values of the orientation angle over 100 Monte Carlo runs for the case with the larger sampling time. T1, T2 and T3 stand for rectangle, triangle and plus, respectively.

	T1		**T2**		**T3**	
	IOU	**RMSE** (*rad*)	**IOU**	**RMSE** (*rad*)	**IOU**	**RMSE** (*rad*)
EKF	0.90	0.09	0.69	1.10	0.79	0.47
VB	0.91	0.07	0.84	0.14	0.88	0.04



(a) (b) (c)

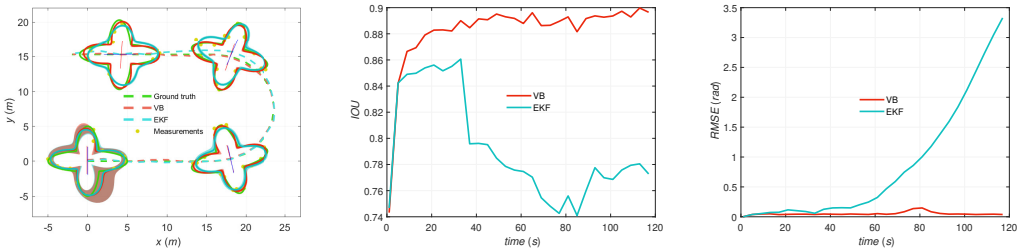
Figure 4.7: (a) displays a typical realization of the simulation for the larger sampling time case with rectangle-shaped object. (b) and (c) show IOU and RMSE for orientation angles over 100 Monte Carlo runs at each instant, respectively.



(a) (b) (c)

Figure 4.8: (a) displays a typical realization of the simulation for the larger sampling time case with triangle-shaped object. (b) and (c) show IOU and RMSE for orientation angles over 100 Monte Carlo runs at each instant, respectively.

In light of the findings of Figures 4.7, 4.8, and 4.9, and also Table 4.4, the measurement update framework based on variational inference surpasses the one with the extended Kalman filter in estimating the extent and the orientation angle.



(a) (b) (c)

Figure 4.9: (a) displays a typical realization of the simulation for the larger sampling time case with plus-shaped object. (b) and (c) show IOU and RMSE for orientation angles over 100 Monte Carlo runs at each instant, respectively.

4.4 Real Data Experiments

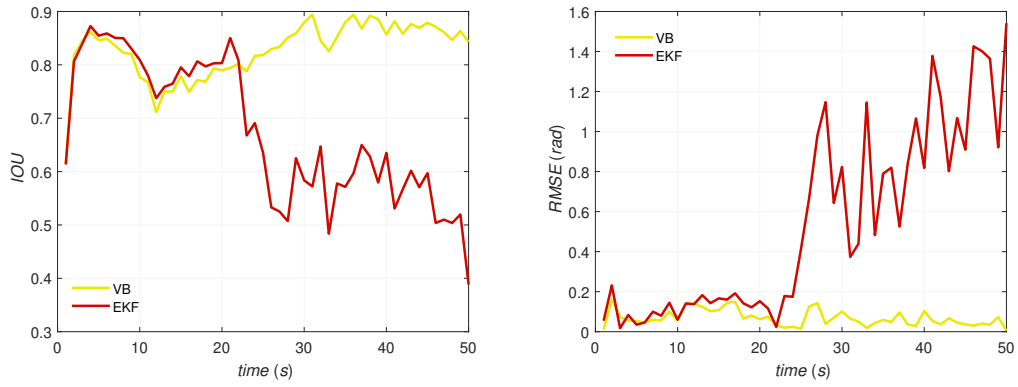
In this experiment, data collection is achieved by a camera installed on an airborne vehicle hovering over a surveillance area. The experiment occurs at a cove on Turkey's western Mediterranean coast, where a dinghy performs agile movements in the scene. Further, the camera captures an image every second throughout the scenario. Extracting measurements from images is a crucial step. At this point, a feature extraction algorithm called Harris corner detector [61] enables us to collect measurements from the object's contour. After providing measurements, both approaches use the same set of parameters listed in Table 4.5.

Table 4.5: Motion and GP parameters used in the real data experiment. P1 represents the motion parameters, and P2 stands for the GP parameters.

	P1				**P2**		
	ΔT	σ_c	σ_ψ	α	σ_r	σ_f	$\boldsymbol{l}$
EKF	1	4	0.2	0.99	0.6	10	$\frac{\pi}{6}$
VB	1	4	0.2	0.99	0.6	10	$\frac{\pi}{6}$

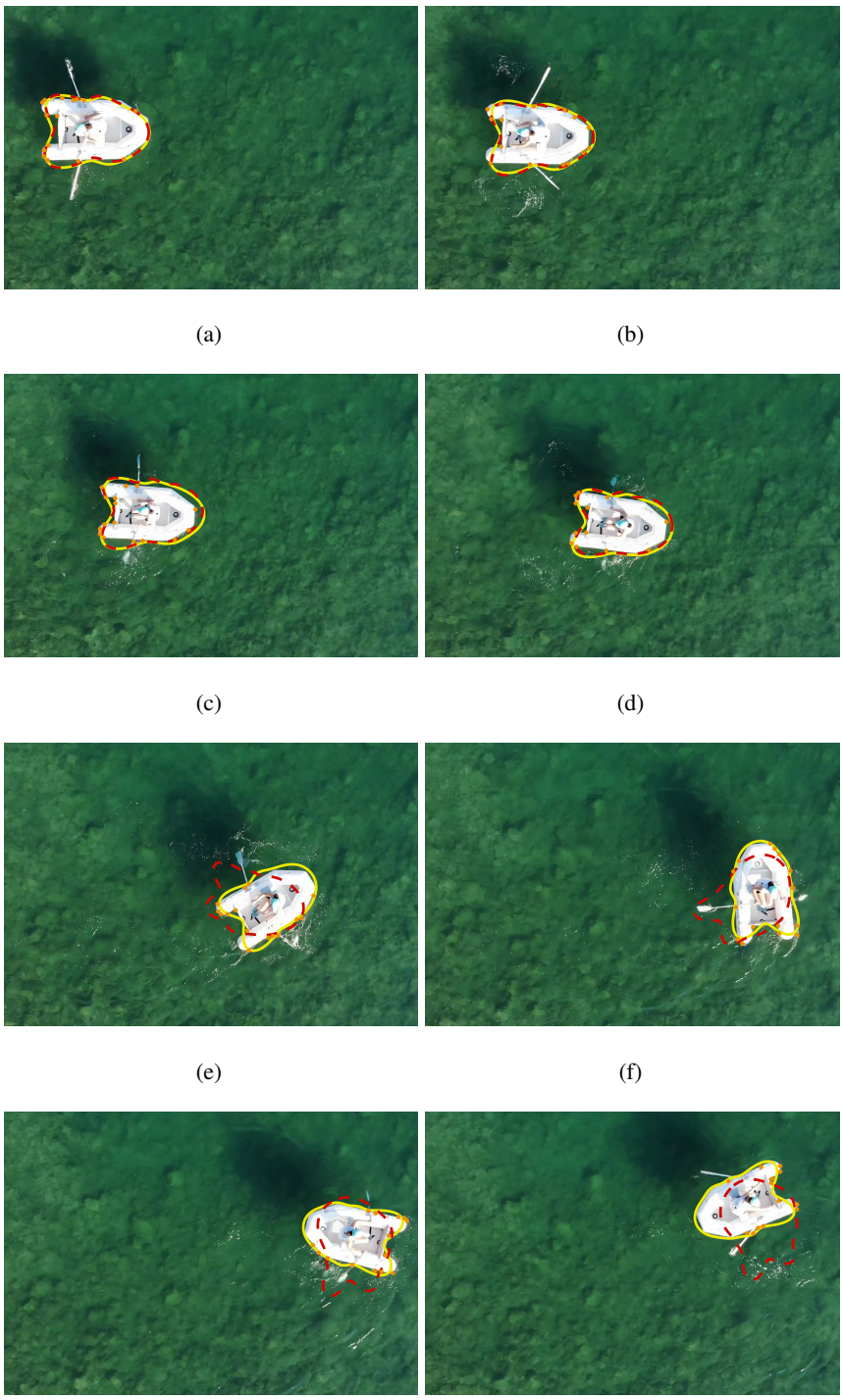
Figure 4.11 visualize the outputs of the algorithms for different instants, in which Figure 4.11(a)-(d) refer to the part where the dinghy movement matches the dynamic model utilized by the algorithms. Both methods perform successfully when

the dinghy moves along a relatively straight line. 4.11(e)-(h), on the other hand, show how the algorithms perform when there is a model mismatch between the dinghy movement and the dynamic model employed by the algorithms. As can be observed, VB performs better in terms of both extent and kinematic estimates once the dinghy begins to maneuver. Furthermore, IOU and RMSE for the orientation angle of the algorithms for each instant are presented in Figure 4.10, which illustrates the algorithms' equivalent performance when the dinghy goes along a relatively straight path and the superiority of VB once the dinghy begins to maneuver.



(a) (b)

Figure 4.10: (a) and (b) display IOU and RMSE for orientation angles at each instant for camera data, respectively.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 4.11: The visualization of the outputs for different instants. (a)-(d) refer to the part where the dinghy movement matches the dynamic model utilized by the algorithms. (e)-(h) display the outputs when there is a model mismatch between the dinghy movement and the dynamic model employed by the algorithms. Yellow line and red dashed line represent the outputs of VB and EKF, respectively. Orange dots stand for measurements.

4.5 Conclusion

In this chapter, an alternative measurement update for an extended target tracking framework is investigated. The variational inference-based alternative measurement update seeks to reduce reliance on the process model's suitability for the scenario and the need for a brief sampling period to obtain accurate estimates. In order to evaluate the performance, simulations and real data experiments are conducted. The variational inference-based measurement update (VB) is compared with the extended Kalman filter-based one. In the case of short sampling time and a match between the scenario and the dynamic model, VB and EKF performances are comparable. On the other hand, VB outperforms EKF if the dynamic model mismatch with the scenario occurs or the sampling time is increased.

CHAPTER 5

CONCLUSION

Multi-sensor fusion combines information from different sensors to enhance knowledge and obtain superior performance. It is a difficult task that requires maintaining the benefits of individual information sources. The fusion performance may be compromised by several challenges, including sensor biases, limited channel capacity, communication channel delays, and the correlation in the acquired information. Another significant challenge is identifying and removing the common information of the different sources, which may stem from the correlation in the gathered information. Otherwise, using the common information more than once in the fusion results in a double counting problem, which may degrade the performance due to overconfidence. The Chernoff fusion is one method that fuses different information sources and aims to prevent the double counting problem.

This thesis focuses on the problem of fusing conflicting information from different sensors. The conventional fusion methods rely on the Gaussian distribution. However, the Gaussian assumption can be restrictive in many applications. Furthermore, such methods do not perform well in the case of outliers and model uncertainties. In this regard, heavy-tailed distributions such as student t-distribution can be a good alternative to Gaussians. With this motivation, we investigate the idea of combining the Chernoff fusion method and student t-distributions. The Chernoff fusion method involves exponentiation and multiplication of the input distributions, which are trivial to perform for Gaussian distributions. On the contrary, it is difficult to obtain the relevant, intermediate expressions for the student t-distributions. In this thesis study, we derived the necessary expressions in order to perform Chernoff fusion with student t-distributions and proposed a novel fusion method. The proposed method obtains

the relevant expressions exactly or approximately, where the approximations are obtained by solving particular optimization problems. The proposed algorithm is called Chernoff fusion with student t-distribution. An extension to CFST, called CFST with memory, is also derived for temporal recursive applications. The difference between the CFST and CFST with memory is the capability of using the fusion output obtained at the previous time instant.

The performance evaluation of the algorithms is a critical task. In this thesis study, we conducted comprehensive experiments to illustrate and test the performance of the methods. These experiments include real data as well as simulations. Real data is collected from different sensors in an autonomous driving environment. In particular, a scenario with a mismatch between one sensor's actual and known characteristics is constructed for both simulations and real data experiments. The generated data is used to compare the performance of the CFST and CFST with memory with various algorithms such as covariance intersection, centralized Kalman filter, and local Kalman filters. The simulations and real data experiments show that the proposed algorithms outperform the alternative, conventional methods in terms of robustness.

Another focus of this thesis is related to an extended target tracking framework constructed with the Gaussian process. This framework's fundamental objective is to track a target's position and extent using noisy measurements from the target contour. The Gaussian process establishes the connection between the measurements and the target extent. The suggested inference methods rely on Kalman filter variations when a nonlinear, implicit measurement model is present. Although such approaches have been shown to produce an effective tracking framework, they have the well-known drawbacks of nonlinear Kalman filtering. For instance, poorly predicted state density and high levels of uncertainty in the model could drastically degrade the estimation performance. The general adoption of the approach in different tracking applications may be limited by these constraints. To address these issues, we proposed a variational approach to improve the estimation performance of the Gaussian process-based extended target tracking methods. The suggested approach is based on a variational Bayes technique for analytical measurement update. The resulting algorithm updates kinematics and extent recursively. It is observed that this recursive filter considerably enhances tracking performance and is robust against model uncertainties via com-

prehensive experiments. In particular, the variational inference-based measurement update performance is compared with the extended Kalman filter-based through simulations and real data experiments. The simulation and real data experiments show that variational measurement update performance is more robust than extended Kalman filter based measurement update for the scenarios with moderate or high levels of uncertainty included in the model.

REFERENCES

[1] J. K. Uhlmann, "General data fusion for estimates with unknown cross covariances," in *Signal Processing, Sensor Fusion, and Target Recognition V* (I. Kadar and V. Libby, eds.), vol. 2755, pp. 536 – 547, International Society for Optics and Photonics, SPIE, 1996.

[2] S. Julier and J. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *Proceedings of the 1997 American Control Conference (Cat. No.97CH36041)*, vol. 4, pp. 2369–2373 vol.4, 1997.

[3] S. Julier and J. Uhlmann, *General Decentralized Data Fusion with Covariance Intersection (CI)*. 06 2001.

[4] N. Wahlström and E. Özkan, "Extended target tracking using Gaussian processes," *IEEE Transactions on Signal Processing*, vol. 63, no. 16, pp. 4165–4178, 2015.

[5] S. Särkkä, *Bayesian Filtering and Smoothing*. USA: Cambridge University Press, 2013.

[6] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 03 1960.

[7] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[8] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Signal processing, sensor fusion, and target recognition VI*, vol. 3068, pp. 182–193, Spie, 1997.

[9] R. Li and S. Nadarajah, "A review of student's t distribution and its generalizations," *Empirical Economics*, vol. 58, no. 3, pp. 1461–1490, 2020.

[10] R. P. S. Mahler, "Optimal/robust distributed data fusion: a unified approach," in *Signal Processing, Sensor Fusion, and Target Recognition IX* (I. Kadar, ed.),

vol. 4052, pp. 128 – 138, International Society for Optics and Photonics, SPIE, 2000.

[11] M. B. Hurley, "An information theoretic justification for covariance intersection and its generalization," in *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002. (IEEE Cat.No.02EX5997)*, vol. 1, pp. 505–511 vol.1, 2002.

[12] M. Uney, D. E. Clark, and S. J. Julier, "Information measures in distributed multitarget tracking," in *14th International Conference on Information Fusion*, pp. 1–8, 2011.

[13] M. F. Huber, "Recursive Gaussian process regression," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3362–3366, IEEE, 2013.

[14] M. F. Huber, "Recursive Gaussian process: On-line regression and learning," *Pattern Recognition Letters*, vol. 45, pp. 85–91, 2014.

[15] S. Grime and H. Durrant-Whyte, "Data fusion in decentralized sensor networks," *Control Engineering Practice*, vol. 2, no. 5, pp. 849–863, 1994.

[16] C.-Y. Chong, "Hierarchical estimation," 1979.

[17] C.-Y. Chong, K.-C. Chang, and S. Mori, "Distributed tracking in distributed sensor networks," in *1986 American Control Conference*, pp. 1863–1868, 1986.

[18] Y. Bar-Shalom, "On the track-to-track correlation problem," *IEEE Transactions on Automatic Control*, vol. 26, no. 2, pp. 571–572, 1981.

[19] Y. Bar-Shalom and L. Campo, "The effect of the common process noise on the two-sensor fused-track covariance," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-22, no. 6, pp. 803–805, 1986.

[20] K. Chang, R. Saha, and Y. Bar-Shalom, "On optimal track-to-track fusion," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 4, pp. 1271–1276, 1997.

[21] V. Shin, Y. Lee, and T.-S. Choi, "Generalized Millman's formula and its application for estimation problems," *Signal Processing*, vol. 86, no. 2, pp. 257–266, 2006.

[22] V. Shin, G. Shevlyakov, and K. Kim, "A new fusion formula and its application to continuous-time linear systems with multisensor environment," *Computational Statistics & Data Analysis*, vol. 52, no. 2, pp. 840–854, 2007.

[23] H. Chen, T. Kirubarajan, and Y. Bar-Shalom, "Performance limits of track-to-track fusion versus centralized estimation: theory and application [sensor fusion]," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 2, pp. 386–400, 2003.

[24] K. Chang, C.-Y. Chong, and S. Mori, "Analytical and computational evaluation of scalable distributed fusion algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 4, pp. 2022–2034, 2010.

[25] D. Nicholson, C. Lloyd, S. Julier, and J. Uhlmann, "Scalable distributed data fusion," in *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002. (IEEE Cat.No.02EX5997)*, vol. 1, pp. 630–635 vol.1, 2002.

[26] F. Bourgault and H. Durrant-Whyte, "Communication in general decentralized filters and the coordinated search strategy," 01 2004.

[27] N. G. Wah and Y. Rong, "Comparison of decentralized tracking algorithms," in *Sixth International Conference of Information Fusion, 2003. Proceedings of the*, vol. 1, pp. 107–113, 2003.

[28] Q. Gan and C. Harris, "Comparison of two measurement fusion methods for Kalman-filter-based multisensor data fusion," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, no. 1, pp. 273–279, 2001.

[29] J. Manyika and H. Durrant-Whyte, "Data fusion and sensor management: A decentralized information-theoretic approach," 1995.

[30] R. Doyle and C. Harris, "Multi-sensor data fusion for helicopter guidance using neuro-fuzzy estimation algorithms," in *1995 IEEE International Conference on*

Systems, Man and Cybernetics. Intelligent Systems for the 21st Century, vol. 2, pp. 1392–1397 vol.2, 1995.

[31] J. Roecker and C. McGillem, "Comparison of two-sensor tracking methods based on state vector fusion and measurement fusion," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 24, no. 4, pp. 447–449, 1988.

[32] Y. Zhu, "Best linear unbiased estimation fusion," 1999.

[33] X. Li, Y. Zhu, and C. Han, "Unified optimal linear estimation fusion. i. Unified models and fusion rules," in *Proceedings of the Third International Conference on Information Fusion*, vol. 1, pp. MOC2/10–MOC2/17 vol.1, 2000.

[34] X. Li and J. Wang, "Unified optimal linear estimation fusion. ii. Discussions and examples," in *Proceedings of the Third International Conference on Information Fusion*, vol. 1, pp. MOC2/18–MOC2/25 vol.1, 2000.

[35] W. Niehsen, "Information fusion based on fast covariance intersection filtering," in *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002. (IEEE Cat.No.02EX5997)*, vol. 2, pp. 901–904 vol.2, 2002.

[36] D. Franken and A. Hupper, "Improved fast covariance intersection for distributed data fusion," in *2005 7th International Conference on Information Fusion*, vol. 1, pp. 7 pp.–, 2005.

[37] C. Chong, "Convex combination and covariance intersection algorithms in distributed fusion," 2001.

[38] D. Clark, S. Julier, R. Mahler, and B. Ristić, "Robust multi-object sensor fusion with unknown correlations," in *Sensor Signal Processing for Defence (SSPD 2010)*, pp. 1–5, 2010.

[39] B. Upcroft, L. Ong, S. Kumar, M. Ridley, T. Bailey, S. Sukkarieh, and H. Durrant-Whyte, "Rich probabilistic representations for bearing only decentralised data fusion," in *2005 7th International Conference on Information Fusion*, vol. 2, pp. 8 pp.–, 2005.

[40] S. J. Julier, "An empirical study into the use of Chernoff information for robust, distributed fusion of Gaussian mixture models," in *2006 9th International Conference on Information Fusion*, pp. 1–8, 2006.

[41] M. Gunay, U. Orguner, and M. Demirekler, "Chernoff fusion of Gaussian mixtures based on sigma-point approximation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 6, pp. 2732–2746, 2016.

[42] M. Roth, E. Özkan, and F. Gustafsson, "A student's t filter for heavy tailed process and measurement noise," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5770–5774, 2013.

[43] H. Zhu, H. Leung, and Z. He, "A variational Bayesian approach to robust sensor fusion based on student-t distribution," *Information Sciences*, vol. 221, pp. 201–214, 2013.

[44] H. Zhu, K. Zou, Y. Li, and H. Leung, "Robust sensor fusion with heavy-tailed noises," *Signal Processing*, vol. 175, p. 107659, 2020.

[45] L. Yan, C. Di, Q. M. J. Wu, and Y. Xia, "Sequential fusion estimation for multi-sensor systems with non-Gaussian noises," *Science China Information Sciences*, vol. 63, 2020.

[46] F. Tronarp, R. Hostettler, and S. Särkkä, "Sigma-point filtering for nonlinear systems with non-additive heavy-tailed noise," in *Proceedings of the 19th International Conference on Information Fusion*, pp. 1859–1866, July 2016.

[47] J. McNamee and F. Stenger, "Construction of fully symmetric numerical integration formulas of fully symmetric numerical integration formulas," *Numerische Mathematik*, vol. 10, pp. 327–344, Nov. 1967.

[48] J. W. Koch, "Bayesian approach to extended object and cluster tracking using random matrices," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 3, pp. 1042–1059, 2008.

[49] M. Feldmann, D. Franken, and W. Koch, "Tracking of extended objects and group targets using random matrices," *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1409–1420, 2010.

[50] K. Granström and U. Orguner, "New prediction for extended targets with random matrices," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 2, pp. 1577–1589, 2014.

[51] J. Lan and X. R. Li, "Tracking of extended object or target group using random matrix—part i: New model and approach," in *2012 15th International Conference on Information Fusion*, pp. 2177–2184, IEEE, 2012.

[52] K. Gilholm and D. Salmond, "Spatial distribution model for tracking extended objects," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 152, no. 5, pp. 364–371, 2005.

[53] K. Gilholm, S. Godsill, S. Maskell, and D. Salmond, "Poisson models for extended target and group tracking," in *Signal and Data Processing of Small Targets 2005*, vol. 5913, p. 59130R, International Society for Optics and Photonics, 2005.

[54] M. Baum and U. D. Hanebeck, "Random hypersurface models for extended object tracking," in *2009 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pp. 178–183, IEEE, 2009.

[55] M. Baum and U. D. Hanebeck, "Shape tracking of extended objects and group targets with star-convex RHMs," in *14th International Conference on Information Fusion*, pp. 1–8, IEEE, 2011.

[56] K. Granström, C. Lundquist, and U. Orguner, "Tracking rectangular and elliptical extended targets using laser measurements," in *14th International Conference on Information Fusion*, pp. 1–8, IEEE, 2011.

[57] K. Granström and C. Lundquist, "On the use of multiple measurement models for extended target tracking," in *Proceedings of the 16th International Conference on Information Fusion*, pp. 1534–1541, IEEE, 2013.

[58] U. Orguner, "A variational measurement update for extended target tracking with random matrices," *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3827–3834, 2012.

[59] S. F. Kara and E. Özkan, "Multi-ellipsoidal extended target tracking using sequential Monte Carlo," in *2018 21st International Conference on Information Fusion (FUSION)*, pp. 1–8, IEEE, 2018.

[60] M. Kumru, H. Köksal, and E. Özkan, "Variational measurement update for extended object tracking using Gaussian processes," *IEEE Signal Processing Letters*, vol. 28, pp. 538–542, 2021.

[61] C. G. Harris, M. Stephens, *et al.*, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, 1988.