# APPROXIMATE METHODS FOR STATE ESTIMATION WITH NONLINEAR MEASUREMENTS AND UNKNOWN NOISE COVARIANCES

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

BY

ERAY LAZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JULY 2023

Approval of the thesis:

**APPROXIMATE METHODS FOR STATE ESTIMATION WITH NONLINEAR MEASUREMENTS AND UNKNOWN NOISE COVARIANCES**

submitted by **ERAY LAZ** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy 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** __________

Prof. Dr. Umut Orguner
Supervisor, **Electrical and Electronics Engineering, METU** __________

**Examining Committee Members:**

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

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

Assist. Prof. Dr. Gökhan Soysal
Electrical and Electronics Engineering, Ankara University __________

Prof. Dr. Murat Efe
Electrical and Electronics Engineering, Ankara University __________

Assist. Prof. Dr. Serkan Sarıtaş
Electrical and Electronics Engineering, METU __________

Date: 04.07.2023

**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: Eray Laz

Signature :

ABSTRACT

APPROXIMATE METHODS FOR STATE ESTIMATION WITH
NONLINEAR MEASUREMENTS AND UNKNOWN NOISE COVARIANCES

Laz, Eray
Ph.D., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Umut Orguner

July 2023, 124 pages

Bayesian state estimation problems with nonlinear measurements and unknown noise covariances are investigated in this thesis.

First, a Gaussian mixture/sum filter in the framework of assumed density filtering is proposed for systems with nonlinear measurement equations. The filter minimizes the Kullback-Leibler divergence from the assumed Gaussian mixture posterior to the true posterior. Since the analytical minimization is not possible, an iterative procedure is developed to obtain the optimal weights, means and covariances of the approximate Gaussian mixture posterior. The resulting Gaussian mixture filter turns out to be a generalization of the (damped) posterior linearization filter to Gaussian mixture posteriors. The performance of the proposed filter is illustrated and compared to alternatives on target tracking examples. The results show that the proposed filter can outperform Gaussian filters as well as the Gaussian sum filter obtaining results very close to a bootstrap particle filter when the number of components in the assumed posterior is sufficiently large.

Second, Bayesian state estimation algorithms are proposed for linear Gaussian systems with the unknown process and measurement noise covariances. The unknown time-varying noise covariances are assumed to be inverse Wishart distributed with Beta-Bartlett transitions. The intractable joint filtered and smoothed posteriors for the state and the noise covariances are calculated by using a scale Gaussian mixture approximation of the Student's *t*-distribution and moment matching. The resulting filters and smoothers are non-iterative unlike the existing solutions in the literature, which brings computational advantages. Furthermore, the proposed filters and smoothers calculate explicit estimates of the noise covariances which might be useful in downstream applications like clutter map formation and/or target classification in radar target tracking. The simulation results show that the proposed algorithms have similar performance as the state of the art solutions while requiring less computational resources.

Keywords: Nonlinear filtering, target tracking, Gaussian mixture, Gaussian sum, Kullback-Leibler divergence, Newton's method, linear systems, Bayesian filtering, Bayesian smoothing, unknown process and measurement noise covariances, inverse Wishart distribution

# ÖZ

## DOĞRUSAL OLMAYAN ÖLÇÜMLER VE BİLİNMEYEN GÜRÜLTÜ KOVARYANSLARI İLE DURUM KESTİRİMİ İÇİN YAKLAŞIK YÖNTEMLER

Laz, Eray
Doktora, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Umut Orguner

Temmuz 2023 , 124 sayfa

Bu tezde, doğrusal olmayan ölçümler ve bilinmeyen gürültü kovaryansları ile Bayes durum kestirim problemleri incelenmektedir.

İlk olarak, doğrusal olmayan ölçüm denklemlerine sahip sistemler için varsayılan yoğunluk süzgeçlemesi çerçevesinde Gauss karışım/toplam süzgeci önerilmektedir. Süzgeç, varsayılan ardıl Gauss karışımından doğru ardıl dağılama olan Kullback-Leibler uzaklığını en aza indirmektedir. Analitik minimizasyon mümkün olmadığından, yaklaşık ardıl Gauss karışımının optimal ağırlıklarını, ortalamalarını ve kovaryanslarını elde etmek için yinelemeli bir prosedür geliştirilmektedir. Ortaya çıkan Gauss karışım filtresi, (sönümlenmiş) ardıl doğrusallaştırma süzgecinin ardıl Gauss karışımına genelleştirilmesi olarak ortaya çıkmaktadır. Önerilen süzgecin performansı, hedef izleme örnekleri üzerinden gösterilmekte ve alternatifleriyle karşılaştırılmaktadır. Sonuçlar, önerilen süzgecin Gauss süzgeçlerinin yanı sıra Gauss toplam süzgecinden daha iyi performans gösterebileceğini ve varsayılan ardıl dağılımdaki bileşen sayısı yeterince büyük olduğunda bootstrap parçacık süzgecine çok yakın sonuçlar

elde ettiğini göstermektedir.

İkinci olarak, bilinmeyen süreç ve ölçüm gürültüsü kovaryansları ile doğrusal Gauss sistemleri için Bayes durum kestirim algoritmaları önerilmektedir. Bilinmeyen zamanla değişen gürültü kovaryanslarının, Beta-Bartlett geçişleriyle ters Wishart dağılıma sahip olduğu varsayılmaktadır. Elde edilmesi zor olan durum ve gürültü kovaryanslarının süzgeçlenmiş ve yumuşatılmış bileşik ardıl dağılımları, Student *t*-dağılımının ölçekli Gauss karışımı yaklaşımı ve moment eşleştirmesi kullanılarak hesaplanmaktadır. Ortaya çıkan süzgeçler ve yumuşatıcılar literatürdeki mevcut çözümlerin aksine yinelemeli değildir. Bu özellik hesaplama avantajları getirmektedir. Ayrıca, önerilen süzgeçler ve yumuşatıcılar gürültü kovaryanslarının açık kestirimlerini hesaplamaktadır. Bu özellik radar hedef izlemede kargaşa haritası oluşumu ve/veya hedef sınıflandırma gibi uygulamalarda faydalı olabilir. Benzetim sonuçları, önerilen algoritmaların daha az hesaplama kaynağı gerektirirken, son teknoloji çözümlerle benzer performansa sahip olduğunu göstermektedir.

Anahtar Kelimeler: Doğrusal olmayan süzgeçleme, hedef takibi, Gauss karışımı, Gauss toplamı, Kullback-Leibler uzaklığı, Newton yöntemi, doğrusal sistemler, Bayes süzgeçleme, Bayes yumuşatma, bilinmeyen süreç ve ölçüm gürültü kovaryansları, ters Wishart dağılımı

To my family

## ACKNOWLEDGMENTS

First and foremost, I would like to express my gratitude to my supervisor Prof. Dr. Umut Orguner. He helped and guided me throughout my thesis studies. He is a kind, hardworking and helpful supervisor. It was a great privilege to work with him.

I would like to thank Assoc. Prof. Dr. Emre Özkan and Assist. Prof. Dr. Gökhan Soysal for being in my thesis monitoring committee. Their comments made this work better. They contributed significant suggestions throughout the entire thesis work.

I would also like to thank the members of my thesis defense committee Prof. Dr. Murat Efe and Assist. Prof. Dr. Serkan Sarıtaş for reviewing my thesis and providing useful comments.

I would like to thank my wife Ayşegül especially for being in my life and for her love, support, patience and encouragement. I also want to thank her for giving birth to our child Batuhan during this thesis period. This cute baby makes my life more meaningful and exciting. I hope that we will have better times to realize our plans after this long journey.

I would like to extend my special thanks to my father Recep, my mother Yasemin and my sisters Emine and Esra for their endless support and encouragement in every step of my life.

I want to thank ASELSAN Inc. for the facilities provided.

# TABLE OF CONTENTS

ABSTRACT . . . . . . . . . . v

ÖZ . . . . . . . . . . vii

ACKNOWLEDGMENTS . . . . . . . . . . x

TABLE OF CONTENTS . . . . . . . . . . xi

LIST OF TABLES . . . . . . . . . . xv

LIST OF FIGURES . . . . . . . . . . xvi

LIST OF ABBREVIATIONS . . . . . . . . . . xix

CHAPTERS

1 INTRODUCTION . . . . . . . . . . 1

1.1 Motivation . . . . . . . . . . 1

1.2 Proposed Methods . . . . . . . . . . 2

1.3 Major Contributions . . . . . . . . . . 3

1.4 The Outline of the Thesis . . . . . . . . . . 3

2 BACKGROUND . . . . . . . . . . 5

2.1 Bayesian Filtering and Smoothing . . . . . . . . . . 5

2.1.1 Bayesian Filtering . . . . . . . . . . 5

2.1.2 Kalman Filter . . . . . . . . . . 7

2.1.3 Bayesian Smoothing . . . . . . . . . . 12

2.1.4 Rauch-Tung-Striebel Smoother . . . . . . . . . . . . . . . . . . 13

2.2 Unscented Transform . . . . . . . . . . . . . . . . . . . . . . . . . 15

2.3 Kullback-Leibler Divergence . . . . . . . . . . . . . . . . . . . . . . 19

2.4 Newton's Method . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

2.5 Inverse Wishart Distribution . . . . . . . . . . . . . . . . . . . . . 25

3 GAUSSIAN MIXTURE FILTERING WITH NONLINEAR MEASUREMENTS MINIMIZING FORWARD KULLBACK-LEIBLER DIVERGENCE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27

3.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27

3.2 Related Literature . . . . . . . . . . . . . . . . . . . . . . . . . . . 29

3.3 Problem Definition . . . . . . . . . . . . . . . . . . . . . . . . . . 31

3.4 Proposed Iterative Measurement Update . . . . . . . . . . . . . . . 32

3.4.1 Implicit Expressions for the Means and Covariances . . . . . . 33

3.4.2 Implicit Expression for the Weights . . . . . . . . . . . . . . . 37

3.4.3 Approximation of the Expectations . . . . . . . . . . . . . . . 38

3.4.4 Iterative Measurement Update . . . . . . . . . . . . . . . . . . 41

3.4.5 Special Case $N = J = 1$ . . . . . . . . . . . . . . . . . . . . 43

3.5 Illustration on a Static Bayesian Estimation Problem . . . . . . . . . 44

3.6 Numerical Results . . . . . . . . . . . . . . . . . . . . . . . . . . . 47

3.6.1 Scenario . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47

3.6.2 Implemented Methods . . . . . . . . . . . . . . . . . . . . . . . 49

3.6.3 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 51

3.7 Conclusions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 52

3.8 Derivations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54

3.8.1 Derivations of (3.9) and (3.21) . . . . . . . . . . . . . . . . . . 54

3.8.1.1 Derivation of the Gradient $\nabla_{\hat{\mathbf{x}}_i} L(q)$ in (3.9a) . . . . . . 54

3.8.1.2 Derivation of the Gradient $\nabla_{\mathbf{\Sigma}_i} L(q)$ in (3.9b) . . . . . . 56

3.8.1.3 Derivation of the Hessian $\nabla_{\hat{\mathbf{x}}_i} \nabla^{\mathrm{T}}_{\hat{\mathbf{x}}_i} L(q)$ in (3.21) . . . . 58

3.8.2 Derivation of (3.24a) . . . . . . . . . . . . . . . . . . . . . . 58

3.8.3 Derivation of (3.56) . . . . . . . . . . . . . . . . . . . . . . . 59

3.8.4 Lemmas Used in the Derivations . . . . . . . . . . . . . . . . . 61

4 BAYESIAN FILTERING/SMOOTHING WITH UNKNOWN NOISE COVARIANCES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 63

4.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . 63

4.2 Problem Definition . . . . . . . . . . . . . . . . . . . . . . . . . 66

4.3 Bayesian Filtering with Unknown PNC . . . . . . . . . . . . . . . 67

4.4 Bayesian Filtering with Unknown MNC . . . . . . . . . . . . . . . 72

4.5 Bayesian Smoothing with Unknown PNC . . . . . . . . . . . . . . 76

4.6 Bayesian Smoothing with Unknown MNC . . . . . . . . . . . . . . 82

4.7 Simulation Results . . . . . . . . . . . . . . . . . . . . . . . . . 83

4.7.1 Bayesian Filtering and Smoothing with Unknown PNC . . . . 83

4.7.2 Bayesian Filtering and Smoothing with Unknown MNC . . . . 89

4.8 Bayesian Filtering and Smoothing when both MNC and PNC are Unknown . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 95

4.9 Conclusions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 97

4.10 Derivations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 99

4.10.1 Derivation of (4.9) . . . . . . . . . . . . . . . . . . . . . . . . 99

4.10.2 Moment Matching for an Infinite Mixture of IW Distributions . 100

4.10.3 Approximation of Joint Predicted Distribution . . . . . . . . . . 102

4.10.4 Calculation of $\mathrm{E}[\mathbf{V}(\mathbf{w}_n)]$ and $\mathrm{E}[\mathbf{V}^2_{mm}(\mathbf{w}_n)]$ for Smoothing with Unknown PNC . . . . . . . . . . . . . . . . . . . . . . . . . 104

5 CONCLUSIONS AND FUTURE WORK . . . . . . . . . . . . . . . . . . 109

REFERENCES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 113

CURRICULUM VITAE . . . . . . . . . . . . . . . . . . . . . . . . . . . . 123

# LIST OF TABLES

TABLES

Table 3.1 Average simulation times of the algorithms in milliseconds. . . . . . 45

Table 4.1 Position and velocity ARMSE of the filters and smoothers for the time-varying PNC scenario. . . . . . . . . . . . . . . . . . . . . . . . . 86

Table 4.2 Position and velocity ARMSE of the filters and smoothers for the constant PNC scenario. . . . . . . . . . . . . . . . . . . . . . . . . . 86

Table 4.3 Bias values for the filtered and smoothed PNC estimates for the time-varying (TV) and constant (C) PNC scenarios. . . . . . . . . . . . 90

Table 4.4 Computation times (per MC run) of the smoothers for the time-varying PNC scenario. . . . . . . . . . . . . . . . . . . . . . . . . . . 90

Table 4.5 Position and velocity ARMSE of the filters and smoothers for the time-varying MNC scenario. . . . . . . . . . . . . . . . . . . . . . . . 92

Table 4.6 Position and velocity ARMSE of the filters and smoothers for the constant MNC scenario. . . . . . . . . . . . . . . . . . . . . . . . . . 92

Table 4.7 Bias values for the filtered and smoothed MNC estimates for the time-varying (TV) and constant (C) MNC scenarios. . . . . . . . . . . . 95

Table 4.8 Computation times (per MC run) of the smoothers for the time-varying MNC scenario. . . . . . . . . . . . . . . . . . . . . . . . . . 96

## LIST OF FIGURES

FIGURES

Figure 2.1 Kalman filter example for two-dimensional target tracking problem. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

Figure 2.2 Kalman smoother example for two-dimensional target tracking problem. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

Figure 2.3 Kalman filter and smoother example for two-dimensional target tracking problem. . . . . . . . . . . . . . . . . . . . . . . . . . . 16

Figure 2.4 Particles and sigma points generated from the distribution given in equation (2.50). $\%99$ confidence ellipses are formed by using particles and sigma points. The weight $\pi_0$ is taken as $0.75$. . . . . . . . . . . 18

Figure 2.5 Particles and sigma points generated from the distribution given in equation (2.50). $\%99$ confidence ellipses are formed by using particles and sigma points. The weight $\pi_0$ is taken as $0.5$. . . . . . . . . . . 19

Figure 2.6 Particles and sigma points propagated through $\mathbf{g}(\mathbf{x})$ given in (2.51). $\%99$ confidence ellipses are formed by using particles and sigma points. . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

Figure 2.7 The distribution $p(\mathbf{x})$ and its approximations found via minimization of forward KL divergence, reverse KL divergence and $\alpha$-divergence. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

Figure 2.8 Geometrical representation of Newton-Raphson method. . . . . 24

Figure 3.1 The estimated posterior densities for the static Bayesian estimation problem. The light blue dots represent the resampled particles of the PF. Gaussian densities are illustrated with 95% confidence ellipses (solid lines) of their covariances centered around their means. Ellipses in dashed lines represent the mean and covariance of the posteriors obtained by PF and FKLF. The weights of the Gaussian components of the posterior obtained by FKLF are written at the left vertex along the semi-major axis of the confidence ellipse of each component. The sensor position is shown with a plus sign in red. . . . . . . . . . . . . . . 46

Figure 3.2 Sensor positions (red plus signs) and 1000 random sample target trajectories used in the simulations. . . . . . . . . . . . . . . . . . . . . 48

Figure 3.3 Position RMSE performance of the filters. . . . . . . . . . . . . 51

Figure 3.4 Velocity RMSE performance of the filters. . . . . . . . . . . . . 52

Figure 3.5 NEES values of the filters. . . . . . . . . . . . . . . . . . . . . . 53

Figure 3.6 Average number of iterations made by the iterative filters. . . . . 54

Figure 4.1 Filtered mean estimates and uncertainties of some entries of the PNC for the proposed method and VB in the time-varying PNC scenario. The estimate uncertainties are shown by the shaded regions representing the interval between the 5th and 95th percentiles over the MC runs. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 87

Figure 4.2 Smoothed mean estimates and uncertainties of some entries of the PNC for the proposed method and VB in the time-varying PNC scenario. The estimate uncertainties are shown by the shaded regions representing the interval between the 5th and 95th percentiles over the MC runs. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 89

Figure 4.3 Filtered mean estimates and uncertainties of some entries of the MNC for the proposed method and VB in the time-varying MNC scenario. The estimate uncertainties are shown by the shaded regions representing the interval between the 5th and 95th percentiles over the MC runs. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93

Figure 4.4 Smoothed mean estimates and uncertainties of some entries of the MNC for the proposed method and VB in the time-varying MNC scenario. The estimate uncertainties are shown by the shaded regions representing the interval between the 5th and 95th percentiles over the MC runs. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94

Figure 4.5 Filtered and smoothed estimates of $\widehat{\mathbf{R}}_n(1,1)$. . . . . . . . . . . . 98

Figure 4.6 Filtered and smoothed estimates of $\widehat{\mathbf{Q}}_n(1,1)$. . . . . . . . . . . . 99

## LIST OF ABBREVIATIONS

| | |
|---|---|
| RTS | Rauch-Tung-Striebel |
| IW | Inverse Wishart |
| KL | Kullback-Leibler |
| RMSE | Root Mean Square Error |
| EKF | Extended Kalman Filter |
| PDF | Probability Density Function |
| MMSE | Minimum Mean Square Error |
| IEKF | Iterated Extended Kalman Filter |
| SLF | Statistically Linearized Filter |
| UKF | Unscented Kalman Filter |
| GHKF | Gauss-Hermite Kalman filter |
| CKF | Cubature Kalman Filter |
| GSF | Gaussian Sum Filter |
| PMF | Point Mass Filter |
| PF | Particle Filter |
| PLF | Posterior Linearization Filter |
| FKLF | Forward Kullback-Leibler Filter |
| EP | Expectation Propagation |
| 2D | 2 Dimensional |
| GPS | Global Positioning System |
| VB | Variational Bayes |
| MNC | Measurement Noise Covariance |
| PNC | Process Noise Covariance |
| OIM | Observed Information Matrix |

| | |
|---|---|
| MAP | Maximum A Posteriori |
| FPI | Fixed Point Iteration |
| TV | Time-Varying |

# CHAPTER 1

# INTRODUCTION

## 1.1 Motivation

Bayesian state estimation problems appear frequently in a variety of applications, including radar target tracking, finance, and navigation. In Bayesian state estimation the aim is to obtain posterior distribution of the states by using the models and the measurements. Since the states change in time and the measurements arrive sequentially, Bayesian state estimation is usually performed recursively in time. The state estimation algorithms processing measurements sequentially as they arrive are called filters. When the estimation problem is linear and Gaussian, the celebrated Kalman filter provides the optimal estimates in the minimum mean square error sense. However, in many practical applications like radar target tracking, the mathematical model of the state and/or the measurements are nonlinear, which makes the estimation problem nonlinear. Nonlinear Bayesian state estimation problems are usually intractable and hence approximations are made to solve them. In the first part of this thesis (the nonlinear) Bayesian state estimation problem is investigated for systems with nonlinear measurements.

In state estimation problems, the state model uncertainties are represented by the process noise while the measurement/sensor uncertainties are modeled with measurement noise. The standard filters/smoothers assume that the statistics of the process and measurement noises are known. For example, in linear Gaussian estimation, a standard assumption is that the covariances of the process and measurement noises are known. On the other hand, in some common applications, such as target tracking, process noise statistics might depend on various factors, like target type, and it might also change in time. Similarly measurement noise characteristics might depend on environmental factors and might be time-varying. As a result a mismatch might appear between the true noise statistics and the statistics used in the filter/smoother, which might reduce the estimation performance. In the second part of this thesis, the problem of linear Gaussian filtering/smoothing with unknown process and measurement noise covariances is studied.

## 1.2 Proposed Methods

The nonlinear filtering problem is solved by minimizing the forward Kullback-Leibler divergence between an approximate distribution and the true posterior distribution. It is assumed that the approximate distribution is a Gaussian mixture. The minimization is performed to find the weights, means and covariances of the components of the Gaussian mixture. When we take the derivatives of the forward KL divergence with respect to the parameters of the unknown Gaussian mixture and equate them to zero, we encounter with implicit equations from which the parameters cannot be obtained explicitly. This problem is solved by using the resulting implicit expressions recursively to converge at their fixed points. In order to speed up convergence, a Newton type recursion is proposed for the mean parameters. The expectations required in the recursions are calculated approximately using unscented transform.

We solve the problem of linear Gaussian filtering/smoothing in a Bayesian framework by modeling the unknown process and measurement noise covariances as inverse Wishart distributed random matrices with Beta-Bartlett transition in time. The filters and smoothers for unknown process noise and measurement noise covariances are derived separately using standard Bayesian formulae, however, many analytical challenges appear due to the intertwined nature of the noise covariances with the states. We use scale Gaussian mixture approximation of Student's $t$-distribution and moment matching to overcome these challenges.

## 1.3 Major Contributions

Below are listed the main contributions of this thesis:

- The solution proposed for nonlinear filtering fills the gap in the literature between the research on Gaussian sum filtering and nonlinear filtering with divergence minimization.
- The solution proposed for nonlinear filtering is the generalization of the (damped) posterior linearization filter to Gaussian mixtures.
- The solution proposed for nonlinear filtering does not use Monte Carlo approximations making it suitable for high dimensional systems.
- The solution proposed for filtering/smoothing with unknown noise covariances is not iterative unlike most of the alternative solutions in the literature, which makes it computationally less demanding with similar performance.
- The solution proposed for filtering/smoothing with unknown noise covariances does not use marginalization and hence calculates the filtered/smoothed estimates of the covariances which can be utilized downstream in data processing, e.g., in target classification based on kinematic information and/or clutter map formation.

## 1.4 The Outline of the Thesis

Chapter 2 presents the necessary background for the subsequent chapters. The Gaussian mixture filtering problem with nonlinear measurements is investigated in Chapter 3. Chapter 4 presents the novel filters/smoothers with unknown noise covariances. The thesis is concluded in Chapter 5.

# CHAPTER 2

# BACKGROUND

In this chapter, we provide the background information necessary for the subsequent chapters. Section 2.1 introduces Bayesian filtering and smoothing along with the closed-form solutions for the linear and Gaussian cases, namely, Kalman filter and Rauch-Tung-Striebel (RTS) smoother. Sections 2.2, 2.3, 2.4, 2.5 give background information on unscented transform, Kullback-Leibler (KL) divergence, Newton's method, and Inverse Wishart (IW) distribution, respectively.

## 2.1 Bayesian Filtering and Smoothing

The general equations for filtering and smoothing are described in this section. In addition, as special cases of the problems, we present the Kalman filter and RTS smoother equations.

### 2.1.1 Bayesian Filtering

In the filtering, we aim to find the posterior distribution of the state $\mathbf{x}_n \in \mathbb{R}^{n_x}$ at time index $n$ given the measurements $\mathbf{y}_{1:n}$ where $\mathbf{y}_n \in \mathbb{R}^{n_y}$, i.e., $p(\mathbf{x}_n|\mathbf{y}_{1:n})$. The time index in $\mathbf{y}_{1:n}$ represents the measurements from time index 1 to time index $n$. This means that in the filtering solution of the state $\mathbf{x}_n$, we use all measurements up to time index $n$. There are two important probability distributions to calculate the filtering distributions. These distributions represent the state transition model and measurement model which are given as

$$\mathbf{x}_n \sim p(\mathbf{x}_n|\mathbf{x}_{n-1}), \tag{2.1a}$$

$$\mathbf{y}_n \sim p(\mathbf{y}_n|\mathbf{x}_n). \tag{2.1b}$$

It is assumed that the model is Markovian. For the Markov state model, the state at time index $n$ is independent of the states and measurements before time index $n-1$ if $\mathbf{x}_{n-1}$ is given. This statement can be expressed mathematically as follows:

$$p(\mathbf{x}_n|\mathbf{x}_{1:n-1}, \mathbf{y}_{1:n-1}) = p(\mathbf{x}_n|\mathbf{x}_{n-1}). \tag{2.2}$$

In addition, the previous state is independent of the future states and measurements if the current state is given. The mathematical expression is given as

$$p(\mathbf{x}_{n-1}|\mathbf{x}_{n:N}, \mathbf{y}_{n:N}) = p(\mathbf{x}_{n-1}|\mathbf{x}_n). \tag{2.3}$$

The last Markov identity that is related to our work is conditional independence of measurements. In this property, the measurement $\mathbf{y}_n$ is independent of the states $\mathbf{x}_{1:n-1}$ and the measurements $\mathbf{y}_{1:n-1}$ if the current state $\mathbf{x}_n$ is given.

$$p(\mathbf{y}_n|\mathbf{x}_{1:n}, \mathbf{y}_{1:n-1}) = p(\mathbf{y}_n|\mathbf{x}_n). \tag{2.4}$$

To obtain the filtered distribution $p(\mathbf{x}_n|\mathbf{y}_{1:n})$, predicted posterior distribution and updated posterior distribution are calculated iteratively. Predicted distribution is obtained by using the Chapman-Kolmogorov equation. Bayes' rule is used for update part of the filtering. Predicted distribution is provided as

$$p(\mathbf{x}_n|\mathbf{y}_{1:n-1}) = \int p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1})\, \mathrm{d}\mathbf{x}_{n-1}. \tag{2.5}$$

This equation can be proven easily. To prove it, we write the joint distribution of states $\mathbf{x}_n$ and $\mathbf{x}_{n-1}$ when $\mathbf{y}_{1:n-1}$ is given. This distribution is written as follows:

$$p(\mathbf{x}_n, \mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}) = p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_{1:n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}) \tag{2.6a}$$

$$= p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}). \tag{2.6b}$$

The second line of the equation is due to the Markov property. Integration is used to obtain the marginal distribution from a joint distribution. If we integrate the joint distribution with respect to $\mathbf{x}_{n-1}$, we obtain predicted distribution in (2.5).

We can now look into the update step. In this step, Bayes' rule is applied to determine the posterior distribution of $\mathbf{x}_n$ as follows:

$$p(\mathbf{x}_n|\mathbf{y}_{1:n}) = \frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{1:n-1})}{\int p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{1:n-1})\, \mathrm{d}\mathbf{x}_n}. \tag{2.7}$$

To prove this formula, we write the distribution of $\mathbf{x}_n$ given $\mathbf{y}_{1:n-1}$ and $\mathbf{y}_n$. Then Bayes' rule is applied as follows:

$$p(\mathbf{x}_n|\mathbf{y}_{1:n}) = \frac{p(\mathbf{y}_n|\mathbf{x}_n, \mathbf{y}_{1:n-1})p(\mathbf{x}_n|\mathbf{y}_{1:n-1})}{\int p(\mathbf{y}_n|\mathbf{x}_n, \mathbf{y}_{1:n-1})p(\mathbf{x}_n|\mathbf{y}_{1:n-1})\, \mathrm{d}\mathbf{x}_n} \tag{2.8a}$$

$$= \frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{1:n-1})}{\int p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{1:n-1})\, \mathrm{d}\mathbf{x}_n}. \tag{2.8b}$$

The second line above is written by using conditional independence of $\mathbf{y}_n$ and $\mathbf{y}_{1:n-1}$ when $\mathbf{x}_n$ is given. As a result, filtering can be performed by recursive evaluation of the following two equations, namely prediction and update equations.

$$p(\mathbf{x}_n|\mathbf{y}_{1:n-1}) = \int p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1})\, \mathrm{d}\mathbf{x}_{n-1}, \tag{2.9a}$$

$$p(\mathbf{x}_n|\mathbf{y}_{1:n}) = \frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{1:n-1})}{\int p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{1:n-1})\, \mathrm{d}\mathbf{x}_n}. \tag{2.9b}$$

### 2.1.2 Kalman Filter

The Kalman filter is investigated in this section. As previously stated, this filter provides closed form solution for the problems including linear Gaussian measurement and dynamic models. Consider the state space representation given as

$$\mathbf{x}_{n+1} = \mathbf{A}_n\mathbf{x}_n + \boldsymbol{\omega}_{n+1}, \qquad \boldsymbol{\omega}_{n+1} \sim \mathcal{N}(\boldsymbol{\omega}_{n+1}; \mathbf{0}, \mathbf{Q}_{n+1}), \tag{2.10a}$$

$$\mathbf{y}_n = \mathbf{C}_n\mathbf{x}_n + \mathbf{v}_n, \qquad \mathbf{v}_n \sim \mathcal{N}(\mathbf{v}_n; \mathbf{0}, \mathbf{R}_n), \tag{2.10b}$$

where the matrices $\mathbf{A}_n$ and $\mathbf{C}_n$ are state transition and measurement matrices, respectively. $\boldsymbol{\omega}_n$ and $\mathbf{v}_n$ show process and measurement noises, respectively. The noises are assumed white and Gaussian distributed with zero mean and covariances $\mathbf{Q}_n$ and $\mathbf{R}_n$. We can represent the state transition model and measurement model for this state space representation as

$$p(\mathbf{x}_n|\mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{x}_n; \mathbf{A}_{n-1}\mathbf{x}_{n-1}, \mathbf{Q}_n), \tag{2.11a}$$

$$p(\mathbf{y}_n|\mathbf{x}_n) = \mathcal{N}(\mathbf{y}_n; \mathbf{C}_n\mathbf{x}_n, \mathbf{R}_n). \tag{2.11b}$$

First, we are going to derive Kalman filter equations analytically. Then, Kalman filter equations will be presented in a concise form. Two important identities are provided in order to obtain the filtering equations. The first identity is on the joint distribution of Gaussian random variables. Assume that random variables $\mathbf{x}$ and $\mathbf{y}$ are Gaussian distributed as follows:

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{\Sigma}), \tag{2.12a}$$

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}; \mathbf{C}\mathbf{x}, \mathbf{R}). \tag{2.12b}$$

The joint distribution of $\mathbf{x}$ and $\mathbf{y}$ can be written as

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x}) \tag{2.13a}$$

$$= \mathcal{N}\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}; \begin{bmatrix}\hat{\mathbf{x}}\\ \mathbf{C}\hat{\mathbf{x}}\end{bmatrix}, \begin{bmatrix}\mathbf{\Sigma} & \mathbf{\Sigma}\mathbf{C}^{\mathrm{T}}\\ \mathbf{C}\mathbf{\Sigma} & \mathbf{C}\mathbf{\Sigma}\mathbf{C}^{\mathrm{T}} + \mathbf{R}\end{bmatrix}\right). \tag{2.13b}$$

From the joint distribution above, the marginal distribution of $\mathbf{y}$ can be obtained as

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}; \mathbf{C}\hat{\mathbf{x}}, \mathbf{C}\mathbf{\Sigma}\mathbf{C}^{\mathrm{T}} + \mathbf{R}). \tag{2.14}$$

In the second identity, we show the conditional distributions for jointly Gaussian random variables $\mathbf{x}$ and $\mathbf{y}$. Assume that the joint distribution of $\mathbf{x}$ and $\mathbf{y}$ is given as

$$p(\mathbf{x}, \mathbf{y}) = \mathcal{N}\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}; \begin{bmatrix}\hat{\mathbf{x}}\\ \hat{\mathbf{y}}\end{bmatrix}, \begin{bmatrix}\mathbf{\Sigma}_{xx} & \mathbf{\Sigma}_{xy}\\ \mathbf{\Sigma}_{xy}^{\mathrm{T}} & \mathbf{\Sigma}_{yy}\end{bmatrix}\right). \tag{2.15}$$

The distributions $p(\mathbf{x})$, $p(\mathbf{y})$, $p(\mathbf{x}|\mathbf{y})$ and $p(\mathbf{y}|\mathbf{x})$ can be written as follows:

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{\Sigma}_{xx}), \tag{2.16a}$$

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}; \hat{\mathbf{y}}, \mathbf{\Sigma}_{yy}), \tag{2.16b}$$

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}} + \mathbf{\Sigma}_{xy}\mathbf{\Sigma}_{yy}^{-1}(\mathbf{y} - \hat{\mathbf{y}}), \mathbf{\Sigma}_{xx} - \mathbf{\Sigma}_{xy}\mathbf{\Sigma}_{yy}^{-1}\mathbf{\Sigma}_{xy}^{\mathrm{T}}), \tag{2.16c}$$

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}; \hat{\mathbf{y}} + \mathbf{\Sigma}_{xy}^{\mathrm{T}}\mathbf{\Sigma}_{xx}^{-1}(\mathbf{x} - \hat{\mathbf{x}}), \mathbf{\Sigma}_{yy} - \mathbf{\Sigma}_{xy}^{\mathrm{T}}\mathbf{\Sigma}_{xx}^{-1}\mathbf{\Sigma}_{xy}). \tag{2.16d}$$

These two identities help us to derive Kalman filter equations easily. We need to find prediction update and measurement update equations which are obtained by finding the means and covariances of the predicted posterior distribution and updated posterior distribution, respectively. While finding these equations, we assume that the prior information for the state is $p(\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0; \hat{\mathbf{x}}_0, \mathbf{P}_0)$. We write the joint distribution of the states $\mathbf{x}_{n-1}$ and $\mathbf{x}_n$ given the measurements $\mathbf{y}_{1:n-1}$ in order to find prediction update equations.

$$p(\mathbf{x}_n, \mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}) = p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}) \tag{2.17a}$$

$$= \mathcal{N}(\mathbf{x}_n; \mathbf{A}_{n-1}\mathbf{x}_{n-1}, \mathbf{Q}_n)\mathcal{N}(\mathbf{x}_{n-1}; \hat{\mathbf{x}}_{n-1}, \mathbf{P}_{n-1}) \tag{2.17b}$$

$$= \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_{n-1} \\ \mathbf{x}_n \end{bmatrix}; \hat{\mathbf{x}}^{\dagger}, \mathbf{P}^{\dagger}\right), \tag{2.17c}$$

where $\hat{\mathbf{x}}^{\dagger}$ and $\mathbf{P}^{\dagger}$ are defined as

$$\hat{\mathbf{x}}^{\dagger} = \begin{bmatrix} \hat{\mathbf{x}}_{n-1} \\ \mathbf{A}_{n-1}\hat{\mathbf{x}}_{n-1} \end{bmatrix}, \tag{2.18a}$$

$$\mathbf{P}^{\dagger} = \begin{bmatrix} \mathbf{P}_{n-1} & \mathbf{P}_{n-1}\mathbf{A}_{n-1}^{\mathrm{T}} \\ \mathbf{A}_{n-1}\mathbf{P}_{n-1} & \mathbf{A}_{n-1}\mathbf{P}_{n-1}\mathbf{A}_{n-1}^{\mathrm{T}} + \mathbf{Q}_n \end{bmatrix}. \tag{2.18b}$$

Equation (2.13) is used to write this joint distribution. The conditional distribution of $\mathbf{x}_n$ given measurements $\mathbf{y}_{1:n-1}$ can be written by utilizing (2.14) as

$$p(\mathbf{x}_n|\mathbf{y}_{1:n-1}) = \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_n^p, \mathbf{P}_n^p) \tag{2.19a}$$

$$= \mathcal{N}(\mathbf{x}_n; \mathbf{A}_{n-1}\hat{\mathbf{x}}_{n-1}, \mathbf{A}_{n-1}\mathbf{P}_{n-1}\mathbf{A}_{n-1}^{\mathrm{T}} + \mathbf{Q}_n) \tag{2.19b}$$

which is the predicted posterior. Updated posterior can be found by using this predicted density. To obtain the updated posterior distribution, we first write the joint distribution of $\mathbf{x}_n$ and $\mathbf{y}_n$ given the measurements $\mathbf{y}_{1:n-1}$.

$$p(\mathbf{x}_n, \mathbf{y}_n|\mathbf{y}_{1:n-1}) = p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{1:n-1}) \tag{2.20a}$$

$$= \mathcal{N}(\mathbf{y}_n; \mathbf{C}_n\mathbf{x}_n, \mathbf{R}_n)\mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_n^p, \mathbf{P}_n^p) \tag{2.20b}$$

$$= \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_n \\ \mathbf{y}_n \end{bmatrix}; \hat{\mathbf{x}}^{\dagger\dagger}, \mathbf{P}^{\dagger\dagger}\right), \tag{2.20c}$$

where the mean $\hat{\mathbf{x}}^{\dagger\dagger}$ and covariance $\mathbf{P}^{\dagger\dagger}$ are provided as

$$\hat{\mathbf{x}}^{\dagger\dagger} = \begin{bmatrix} \hat{\mathbf{x}}_n^p \\ \mathbf{C}_n\hat{\mathbf{x}}_n^p \end{bmatrix}, \tag{2.21a}$$

$$\mathbf{P}^{\dagger\dagger} = \begin{bmatrix} \mathbf{P}_n^p & \mathbf{P}_n^p\mathbf{C}_n^{\mathrm{T}} \\ \mathbf{C}_n\mathbf{P}_n^p & \mathbf{C}_n\mathbf{P}_n^p\mathbf{C}_n^{\mathrm{T}} + \mathbf{R}_n \end{bmatrix}. \tag{2.21b}$$

The last equation is written by using (2.13). Conditional distribution $p(\mathbf{x}_n|\mathbf{y}_n, \mathbf{y}_{1:n-1})$ can be written by utilizing (2.16).

$$p(\mathbf{x}_n|\mathbf{y}_n, \mathbf{y}_{1:n-1}) = p(\mathbf{x}_n|\mathbf{y}_{1:n}) \tag{2.22}$$

$$= \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_n, \mathbf{P}_n), \tag{2.23}$$

where

$$\hat{\mathbf{x}}_n = \hat{\mathbf{x}}_n^p + \mathbf{P}_n^p \mathbf{C}_n^{\mathrm{T}} \left( \mathbf{C}_n \mathbf{P}_n^p \mathbf{C}_n^{\mathrm{T}} + \mathbf{R}_n \right)^{-1} (\mathbf{y}_n - \mathbf{C}_n \hat{\mathbf{x}}_n^p), \tag{2.24a}$$

$$\mathbf{P}_n = \mathbf{P}_n^p - \mathbf{P}_n^p \mathbf{C}_n^{\mathrm{T}} \left( \mathbf{C}_n \mathbf{P}_n^p \mathbf{C}_n^{\mathrm{T}} + \mathbf{R}_n \right)^{-1} \mathbf{C}_n \mathbf{P}_n^p. \tag{2.24b}$$

For the sake of brevity, we can write update equations using innovation covariance $\mathbf{S}_n$ and Kalman gain $\mathbf{K}_n$. These two parameters are defined as

$$\mathbf{S}_n = \mathbf{C}_n \mathbf{P}_n^p \mathbf{C}_n^{\mathrm{T}} + \mathbf{R}_n, \tag{2.25a}$$

$$\mathbf{K}_n = \mathbf{P}_n^p \mathbf{C}_n^{\mathrm{T}} \mathbf{S}_n^{-1}. \tag{2.25b}$$

By using these parameters, we can express Kalman filter equations compactly as follows:

$$\hat{\mathbf{x}}_n^p = \mathbf{A}_{n-1} \hat{\mathbf{x}}_{n-1}, \tag{2.26a}$$

$$\mathbf{P}_n^p = \mathbf{A}_{n-1} \mathbf{P}_{n-1} \mathbf{A}_{n-1}^{\mathrm{T}} + \mathbf{Q}_n, \tag{2.26b}$$

$$\hat{\mathbf{x}}_n = \hat{\mathbf{x}}_n^p + \mathbf{K}_n (\mathbf{y}_n - \mathbf{C}_n \hat{\mathbf{x}}_n^p), \tag{2.26c}$$

$$\mathbf{P}_n = \mathbf{P}_n^p - \mathbf{K}_n \mathbf{S}_n \mathbf{K}_n^{\mathrm{T}}. \tag{2.26d}$$

The first two lines of (2.26) form the prediction update part. The remaining two lines form the measurement update part. In order to run Kalman filter, we start with an initial state whose distribution is given as $p(\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0; \hat{\mathbf{x}}_0, \mathbf{P}_0)$. Then we evaluate (2.26) recursively.

**EXAMPLE:** We perform a two-dimensional target tracking simulation. In this simulation, we have a four-dimensional state consisting of positions and velocities in $x$ and $y$ directions. The state is represented as $\mathbf{x} = [x, \dot{x}, y, \dot{y}]^{\mathrm{T}}$ where $x$ and $y$ show positions while $\dot{x}$ and $\dot{y}$ show velocities. The state space representation of the problem is provided as

$$\mathbf{x}_{n+1} = \mathbf{A}\mathbf{x}_n + \boldsymbol{\omega}_{n+1}, \qquad \boldsymbol{\omega}_{n+1} \sim \mathcal{N}(\boldsymbol{\omega}_{n+1}; \mathbf{0}, \mathbf{Q}), \tag{2.27a}$$

$$\mathbf{y}_n = \mathbf{C}\mathbf{x}_n + \mathbf{v}_n, \qquad \mathbf{v}_n \sim \mathcal{N}(\mathbf{v}_n; \mathbf{0}, \mathbf{R}). \tag{2.27b}$$

The state transition matrix and process noise covariance matrix are defined as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_d \end{bmatrix}, \qquad \mathbf{A}_d = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \tag{2.28a}$$

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_d \end{bmatrix}, \qquad \mathbf{Q}_d = q \begin{bmatrix} \Delta t^3/3 & \Delta t^2/2 \\ \Delta t^2/2 & \Delta t \end{bmatrix}, \tag{2.28b}$$

where $q$ shows process noise's spectral density. Measurements are taken with time interval $\Delta t$. The matrices related to the measurement model, i.e. $\mathbf{C}$ and $\mathbf{R}$, are given as

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{2.29a}$$

$$\mathbf{R} = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}, \tag{2.29b}$$

where $\sigma_1^2$ and $\sigma_2^2$ show measurement noise variances in $x$ and $y$ directions, respectively. The sensor, as shown in the measurement model, measures the target's location in both the $x$ and $y$ directions.

In the simulations, we choose the parameters $\sigma_1^2 = \sigma_2^2 = 1$ m$^2$ and $q = 2$ m$^2$/s$^3$. Additionally, the time interval is chosen as $\Delta t = 0.2$ s. The parameters of the initial state distribution are chosen as $\hat{\mathbf{x}}_0 = [0, 0, 0, 0]^{\mathrm{T}}$ and $\mathbf{P}_0 = \mathrm{diag}([4, 1, 4, 1])$. The position estimates of the Kalman filter are shown in Figure 2.1.

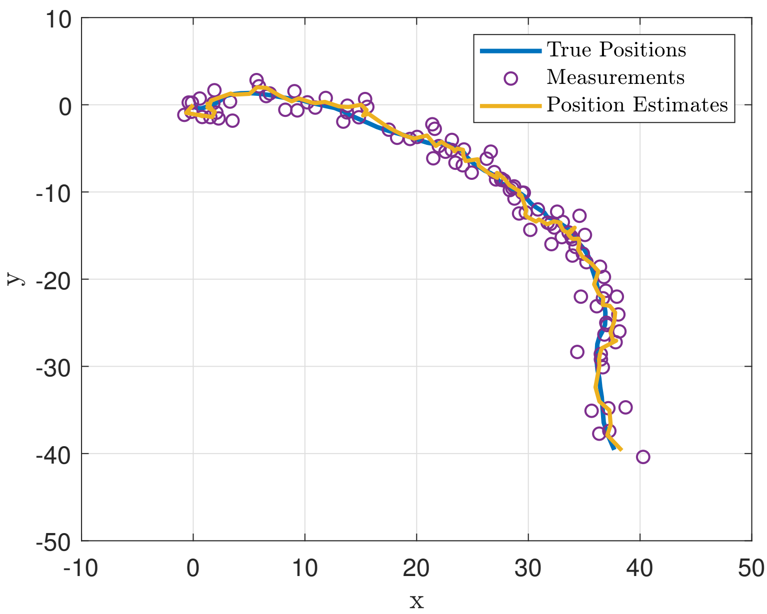



Figure 2.1: Kalman filter example for two-dimensional target tracking problem.

When Figure 2.1 is examined, it is realized that state estimates are close to the true trajectory. This indicates that the Kalman filter reduces position error when compared to measurements. To support this expression numerically, we look at the position root mean square error (RMSE) of estimates and measurements. Position RMSE values are given as $0.83$ m and $1.44$ m for estimates and measurements, respectively.

### 2.1.3 Bayesian Smoothing

The posterior distribution that we need to find in smoothing is $p(\mathbf{x}_n|\mathbf{y}_{1:N})$ where $n \leq N$. In order to find smoothed distribution, we write the joint distribution of $\mathbf{x}_{n+1}$ and $\mathbf{x}_n$ given $\mathbf{y}_{1:N}$.

$$\begin{aligned} p(\mathbf{x}_n, \mathbf{x}_{n+1}|\mathbf{y}_{1:N}) &= p(\mathbf{x}_n|\mathbf{x}_{n+1}, \mathbf{y}_{1:N})p(\mathbf{x}_{n+1}|\mathbf{y}_{1:N}) && \text{(2.30a)} \\ &= p(\mathbf{x}_n|\mathbf{x}_{n+1}, \mathbf{y}_{1:n})p(\mathbf{x}_{n+1}|\mathbf{y}_{1:N}) && \text{(2.30b)} \\ &= \frac{p(\mathbf{x}_n, \mathbf{x}_{n+1}|\mathbf{y}_{1:n})}{p(\mathbf{x}_{n+1}|\mathbf{y}_{1:n})}p(\mathbf{x}_{n+1}|\mathbf{y}_{1:N}) && \text{(2.30c)} \\ &= \frac{p(\mathbf{x}_{n+1}|\mathbf{x}_n, \mathbf{y}_{1:n})p(\mathbf{x}_n|\mathbf{y}_{1:n})}{p(\mathbf{x}_{n+1}|\mathbf{y}_{1:n})}p(\mathbf{x}_{n+1}|\mathbf{y}_{1:N}) && \text{(2.30d)} \\ &= \frac{p(\mathbf{x}_{n+1}|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{1:n})}{p(\mathbf{x}_{n+1}|\mathbf{y}_{1:n})}p(\mathbf{x}_{n+1}|\mathbf{y}_{1:N}). && \text{(2.30e)} \end{aligned}$$

The Markov property is used to write the second and fifth lines of the equation. The rest of the lines employ Bayes' rule. In order to obtain the posterior distribution of $\mathbf{x}_n$ given $\mathbf{y}_{1:N}$, we need to integrate the joint distribution with respect to $\mathbf{x}_{n+1}$.

$$\begin{aligned} p(\mathbf{x}_n|\mathbf{y}_{1:N}) &= \int p(\mathbf{x}_n, \mathbf{x}_{n+1}|\mathbf{y}_{1:N}) \, \mathrm{d}\mathbf{x}_{n+1} && \text{(2.31a)} \\ &= \int \frac{p(\mathbf{x}_{n+1}|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{1:n})}{p(\mathbf{x}_{n+1}|\mathbf{y}_{1:n})}p(\mathbf{x}_{n+1}|\mathbf{y}_{1:N}) \, \mathrm{d}\mathbf{x}_{n+1}. && \text{(2.31b)} \end{aligned}$$

The smoothed distribution can be written more compactly as

$$p(\mathbf{x}_n|\mathbf{y}_{1:N}) = p(\mathbf{x}_n|\mathbf{y}_{1:n}) \int \frac{p(\mathbf{x}_{n+1}|\mathbf{x}_n)p(\mathbf{x}_{n+1}|\mathbf{y}_{1:N})}{p(\mathbf{x}_{n+1}|\mathbf{y}_{1:n})} \, \mathrm{d}\mathbf{x}_{n+1}. \tag{2.32}$$

We need to use (2.32) to obtain smoothed distribution. In this equation, we need state transition distribution, filtered distribution, predicted distribution and lastly smoothed distribution at the next time step.

### 2.1.4 Rauch-Tung-Striebel Smoother

In this part, RTS smoother is derived for linear and Gaussian problems. Consider the state space representation given in (2.10). In the derivation, we try to find the parameters of the following smoothed distribution.

$$p(\mathbf{x}_n|\mathbf{y}_{1:N}) = \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_n^s, \mathbf{P}_n^s). \tag{2.33}$$

The first step that we perform is to obtain joint distribution of $\mathbf{x}_{n+1}$ and $\mathbf{x}_n$ given measurements $\mathbf{y}_{1:n}$.

$$\begin{aligned} p(\mathbf{x}_n, \mathbf{x}_{n+1}|\mathbf{y}_{1:n}) &= p(\mathbf{x}_{n+1}|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{1:n}) & \text{(2.34a)} \\ &= \mathcal{N}(\mathbf{x}_{n+1}; \mathbf{A}_n\mathbf{x}_n, \mathbf{Q}_{n+1})\mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_n, \mathbf{P}_n) & \text{(2.34b)} \\ &= \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_n \\ \mathbf{x}_{n+1} \end{bmatrix}; \hat{\mathbf{x}}_1^\dagger, \mathbf{P}_1^\dagger\right), & \text{(2.34c)} \end{aligned}$$

where the mean $\hat{\mathbf{x}}_1^\dagger$ and covariance $\mathbf{P}_1^\dagger$ are given as

$$\hat{\mathbf{x}}_1^\dagger = \begin{bmatrix} \hat{\mathbf{x}}_n \\ \mathbf{A}_n\hat{\mathbf{x}}_n \end{bmatrix}, \tag{2.35a}$$

$$\mathbf{P}_1^\dagger = \begin{bmatrix} \mathbf{P}_n & \mathbf{P}_n\mathbf{A}_n^{\mathrm{T}} \\ \mathbf{A}_n\mathbf{P}_n & \mathbf{A}_n\mathbf{P}_n\mathbf{A}_n^{\mathrm{T}} + \mathbf{Q}_{n+1} \end{bmatrix}. \tag{2.35b}$$

Note that this equation has been written by using (2.13). Now, we write the distribution of $\mathbf{x}_n$ given $\mathbf{x}_{n+1}$ and all measurements $\mathbf{y}_{1:N}$.

$$\begin{aligned} p(\mathbf{x}_n|\mathbf{x}_{n+1}, \mathbf{y}_{1:N}) &= p(\mathbf{x}_n|\mathbf{x}_{n+1}, \mathbf{y}_{1:n}) & \text{(2.36a)} \\ &= \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_2^\dagger, \mathbf{P}_2^\dagger). & \text{(2.36b)} \end{aligned}$$

The Markov property is used in the first line above. The parameters $\hat{\mathbf{x}}_2^\dagger$ and $\mathbf{P}_2^\dagger$ are found by using (2.16).

$$\hat{\mathbf{x}}_2^\dagger = \hat{\mathbf{x}}_n + \mathbf{P}_n\mathbf{A}_n^{\mathrm{T}}\left(\mathbf{A}_n\mathbf{P}_n\mathbf{A}_n^{\mathrm{T}} + \mathbf{Q}_{n+1}\right)^{-1}(\mathbf{x}_{n+1} - \mathbf{A}_n\hat{\mathbf{x}}_n), \tag{2.37a}$$

$$\mathbf{P}_2^\dagger = \mathbf{P}_n - \mathbf{P}_n\mathbf{A}_n^{\mathrm{T}}\left(\mathbf{A}_n\mathbf{P}_n\mathbf{A}_n^{\mathrm{T}} + \mathbf{Q}_{n+1}\right)^{-1}\mathbf{A}_n\mathbf{P}_n. \tag{2.37b}$$

A matrix $\mathbf{G}_n$ is defined to express the last equation compactly.

$$\mathbf{G}_n = \mathbf{P}_n\mathbf{A}_n^{\mathrm{T}}\left(\mathbf{A}_n\mathbf{P}_n\mathbf{A}_n^{\mathrm{T}} + \mathbf{Q}_{n+1}\right)^{-1}. \tag{2.38}$$

$\hat{\mathbf{x}}_2^\dagger$ and $\mathbf{P}_2^\dagger$ are written by using $\mathbf{G}_n$ as

$$\hat{\mathbf{x}}_2^\dagger = \hat{\mathbf{x}}_n + \mathbf{G}_n(\mathbf{x}_{n+1} - \mathbf{A}_n\hat{\mathbf{x}}_n), \tag{2.39a}$$

$$\mathbf{P}_2^\dagger = \mathbf{P}_n - \mathbf{G}_n\left(\mathbf{A}_n\mathbf{P}_n\mathbf{A}_n^{\mathrm{T}} + \mathbf{Q}_{n+1}\right)\mathbf{G}_n^{\mathrm{T}}. \tag{2.39b}$$

We can obtain the distribution $p(\mathbf{x}_n, \mathbf{x}_{n+1}|\mathbf{y}_{1:N})$ simply by multiplying the distributions $p(\mathbf{x}_n|\mathbf{x}_{n+1}, \mathbf{y}_{1:N})$ and $p(\mathbf{x}_{n+1}|\mathbf{y}_{1:N})$.

$$p(\mathbf{x}_n, \mathbf{x}_{n+1}|\mathbf{y}_{1:N}) = p(\mathbf{x}_n|\mathbf{x}_{n+1}, \mathbf{y}_{1:N})p(\mathbf{x}_{n+1}|\mathbf{y}_{1:N}) \tag{2.40a}$$

$$= \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_2^\dagger, \mathbf{P}_2^\dagger)\mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1}^s, \mathbf{P}_{n+1}^s) \tag{2.40b}$$

$$= \mathcal{N}\left(\begin{bmatrix}\mathbf{x}_{n+1}\\ \mathbf{x}_n\end{bmatrix}; \hat{\mathbf{x}}_3^\dagger, \mathbf{P}_3^\dagger\right), \tag{2.40c}$$

where the quantities $\hat{\mathbf{x}}_3^\dagger$ and $\mathbf{P}_3^\dagger$ are given as

$$\hat{\mathbf{x}}_3^\dagger = \begin{bmatrix}\hat{\mathbf{x}}_{n+1}^s\\ \hat{\mathbf{x}}_n + \mathbf{G}_n(\hat{\mathbf{x}}_{n+1}^s - \mathbf{A}_n\hat{\mathbf{x}}_n)\end{bmatrix}, \tag{2.41a}$$

$$\mathbf{P}_3^\dagger = \begin{bmatrix}\mathbf{P}_{n+1}^s & \mathbf{P}_{n+1}^s\mathbf{G}_n^{\mathrm{T}}\\ \mathbf{G}_n\mathbf{P}_{n+1}^s & \mathbf{G}_n\mathbf{P}_{n+1}^s\mathbf{G}_n^{\mathrm{T}} + \mathbf{P}_2^\dagger\end{bmatrix}. \tag{2.41b}$$

The parameters of the smoothed distribution can be written by utilizing (2.16) as

$$\hat{\mathbf{x}}_n^s = \hat{\mathbf{x}}_n + \mathbf{G}_n(\hat{\mathbf{x}}_{n+1}^s - \mathbf{A}_n\hat{\mathbf{x}}_n), \tag{2.42a}$$

$$\mathbf{P}_n^s = \mathbf{G}_n\mathbf{P}_{n+1}^s\mathbf{G}_n^{\mathrm{T}} + \mathbf{P}_2^\dagger \tag{2.42b}$$

$$= \mathbf{P}_n + \mathbf{G}_n\left(\mathbf{P}_{n+1}^s - \mathbf{A}_n\mathbf{P}_n\mathbf{A}_n^{\mathrm{T}} - \mathbf{Q}_{n+1}\right)\mathbf{G}_n^{\mathrm{T}}. \tag{2.42c}$$

As a result, we can express smoothing equations compactly as follows:

$$\hat{\mathbf{x}}_{n+1}^p = \mathbf{A}_n\hat{\mathbf{x}}_n, \tag{2.43a}$$

$$\mathbf{P}_{n+1}^p = \mathbf{A}_n\mathbf{P}_n\mathbf{A}_n^{\mathrm{T}} + \mathbf{Q}_{n+1}, \tag{2.43b}$$

$$\hat{\mathbf{x}}_n^s = \hat{\mathbf{x}}_n + \mathbf{G}_n(\hat{\mathbf{x}}_{n+1}^s - \hat{\mathbf{x}}_{n+1}^p), \tag{2.43c}$$

$$\mathbf{P}_n^s = \mathbf{P}_n + \mathbf{G}_n\left(\mathbf{P}_{n+1}^s - \mathbf{P}_{n+1}^p\right)\mathbf{G}_n^{\mathrm{T}}. \tag{2.43d}$$

**EXAMPLE:** In order to see the results of applying smoothing for a problem, we use the same problem used in Section 2.1.2. Note that smoothing iterations start from the end of the filtering. This means that $\hat{\mathbf{x}}_N^s = \hat{\mathbf{x}}_N$. The position estimates of the smoother

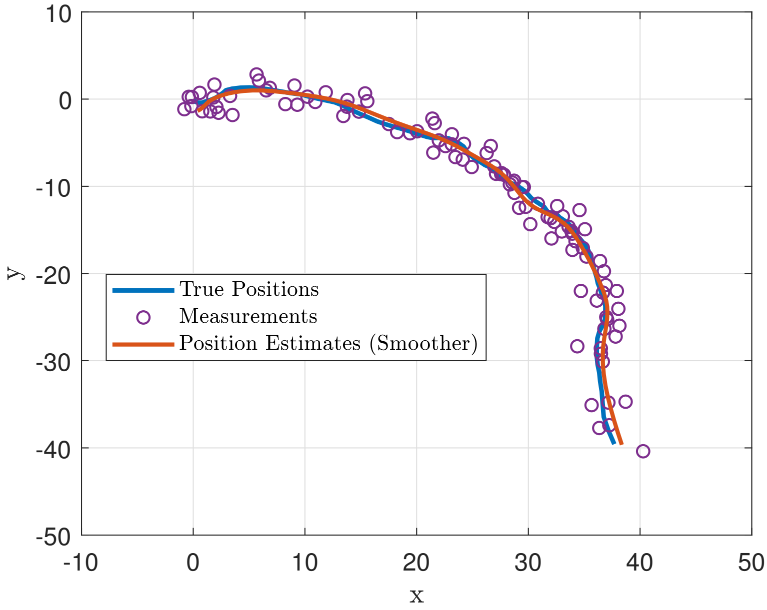



Figure 2.2: Kalman smoother example for two-dimensional target tracking problem.

are shown in Figure 2.2. We can see from Figure 2.2 that the smoother estimates are very close to the true position. Position RMSE value for smoother is found as $0.52$ m. This value was found as $0.83$ m for the filtering solution. According to this example, we can say that the smoother outperforms the filter. In addition, we need to emphasize that there are oscillations in the filter estimates. These oscillations are reduced by the smoother. To visualize this effect, both the Kalman filter and smoother position estimates are plotted in Figure 2.3.

## 2.2 Unscented Transform

In this thesis, we will be interested in calculating $\mathrm{E}[\mathbf{g}(\mathbf{x})]$ where $\mathbf{g}(\cdot)$ is an arbitrary, in general, nonlinear function. Suppose that $\mathbf{y} = \mathbf{g}(\mathbf{x})$ and $\mathbf{x} \sim \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{\Sigma}_x)$, the mean of $\mathbf{g}(\mathbf{x})$, i.e., $\hat{\mathbf{y}} = \mathrm{E}[\mathbf{g}(\mathbf{x})]$, is calculated as

$$\hat{\mathbf{y}} = \int \mathbf{g}(\mathbf{x}) \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{\Sigma}_x) \, \mathrm{d}\mathbf{x}. \tag{2.44}$$

Analytically, taking the integration in (2.44) is not possible. We need to use numerical methods to evaluate the integral expressions. One of these numerical approaches is to use sigma points [1]. The authors in [1] state that an arbitrary nonlinear func-

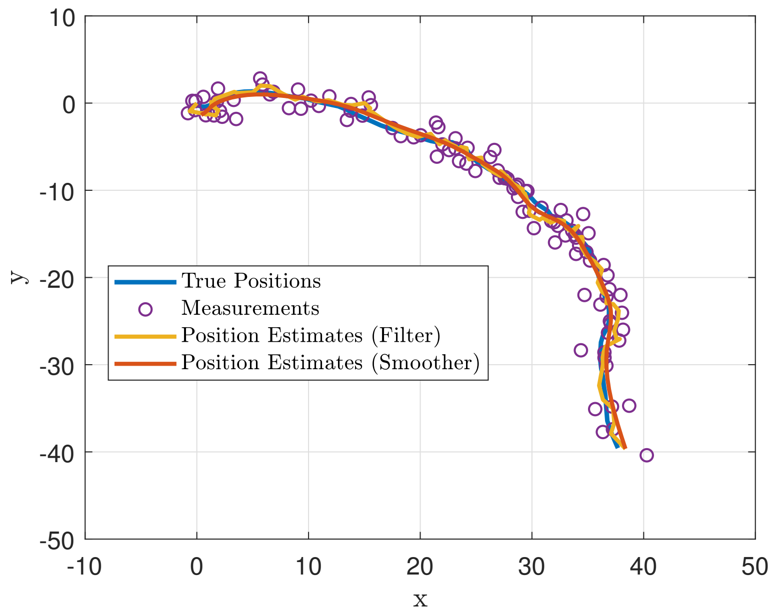



Figure 2.3: Kalman filter and smoother example for two-dimensional target tracking problem.

tion can be more difficult to approximate than a probability distribution. Extended Kalman filter (EKF) can be given as an example for this statement. In EKF, the nonlinear function is linearized by using Taylor series expansion. This is an error-prone process. Instead of this kind of approach, one can generate samples from the related distribution and propagate these samples through the nonlinearities.

In unscented transform, sigma points are generated from a distribution having mean $\hat{\mathbf{x}}$ and covariance $\mathbf{\Sigma}_x$. The sigma points are chosen such that their weighted mean and covariance are the same as those of the original distribution. The placements of the sigma points vary when the weights of the sigma points are changed. The weights can be chosen such that higher order information of the original distribution can be attained with the sigma points. The sigma points for the distribution $\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{\Sigma}_x)$ are chosen as

$$\mathbf{x}_0 = \hat{\mathbf{x}}, \tag{2.45a}$$

$$\mathbf{x}_i = \hat{\mathbf{x}} + \left[\sqrt{\frac{n_x}{1-\pi_0}\Sigma_x}\right]_{:,i}, \tag{2.45b}$$

$$\mathbf{x}_{i+n_x} = \hat{\mathbf{x}} - \left[\sqrt{\frac{n_x}{1-\pi_0}\Sigma_x}\right]_{:,i}, \tag{2.45c}$$

for $i = 1, ..., n_x$ where $\pi_0$ is the weight that is assigned to the sigma point which is at the mean. $\sqrt{\cdot}$ represents the square root of the matrix. In addition, $[]_{:,i}$ denotes the $i^{th}$ column of the matrix. As can be realized, we have $2n_x + 1$ sigma points. The weights of the sigma points are specified as

$$\pi_0 = \pi_0, \tag{2.46a}$$

$$\pi_i = \frac{1-\pi_0}{2n_x}, \tag{2.46b}$$

$$\pi_{i+n_x} = \frac{1-\pi_0}{2n_x}, \tag{2.46c}$$

for $i = 1, ..., n_x$. Note that the summation of the weights is equal to 1. The weights are chosen equal except for the sigma point at the mean. However, $\pi_0$ can be chosen such that all weights are equal. Examining both sigma points and weights formulations reveals that the sigma points are chosen deterministically and the mean and covariance of the sigma points are given as

$$\sum_{i=0}^{2n_x} \pi_i \mathbf{x}_i = \hat{\mathbf{x}}, \tag{2.47a}$$

$$\sum_{i=0}^{2n_x} \pi_i (\mathbf{x}_i - \hat{\mathbf{x}})(\mathbf{x}_i - \hat{\mathbf{x}})^{\mathrm{T}} = \boldsymbol{\Sigma}_x. \tag{2.47b}$$

The sigma points are propagated through the nonlinear function $\mathbf{y} = \mathbf{g}(\mathbf{x})$.

$$\mathbf{y}_i = \mathbf{g}(\mathbf{x}_i) \tag{2.48}$$

for $i = 0, ..., 2n_x$. Then, we find mean $\hat{\mathbf{y}}$ and covariance $\boldsymbol{\Sigma}_y$ of $\mathbf{y}$ as

$$\hat{\mathbf{y}} = \sum_{i=0}^{2n_x} \pi_i \mathbf{y}_i, \tag{2.49a}$$

$$\boldsymbol{\Sigma}_y = \sum_{i=0}^{2n_x} \pi_i (\mathbf{y}_i - \hat{\mathbf{y}})(\mathbf{y}_i - \hat{\mathbf{y}})^{\mathrm{T}}. \tag{2.49b}$$

**EXAMPLE:** In order to make the concept of unscented transformation clearer, a simple graphic illustration is provided. The vector $\mathbf{x}$ is taken such that it represents the position in $x$ and $y$ directions, i.e. $\mathbf{x} = [x, y]^{\mathrm{T}}$. The distribution of $\mathbf{x}$ is given as

$$\mathbf{x} \sim \mathcal{N}\left(\mathbf{x}; \begin{bmatrix} 10 \\ 10 \end{bmatrix}, \begin{bmatrix} 4 & 2 \\ 2 & 8 \end{bmatrix}\right). \tag{2.50}$$

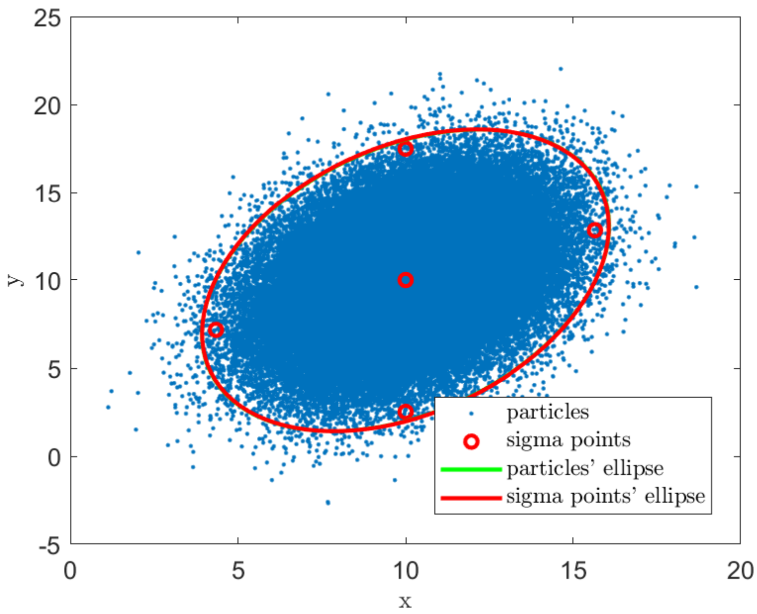



Figure 2.4: Particles and sigma points generated from the distribution given in equation (2.50). $\%99$ confidence ellipses are formed by using particles and sigma points. The weight $\pi_0$ is taken as $0.75$.

Sigma points are generated from this distribution by choosing $\pi_0 = 0.75$. The outcome is displayed in Figure 2.4. The particles in Figure 2.4 are generated randomly from the distribution given in (2.50). Note that we have $5$ sigma points because the dimension of the vector $\mathbf{x}$ is $2$. In addition, the covariances obtained by using particles and sigma points are equal as expected. In this figure, we have chosen $\pi_0 = 0.75$. If we choose it different from $0.75$, then the positions of the sigma points change. However, the same mean and covariance are obtained again. When we set $\pi_0 = 0.5$, we get Figure 2.5. The sigma points given in (2.45) depend on $\pi_0$. If the value of $\pi_0$ is increased, sigma points move away from the mean and vice versa as observed in Figures 2.4 and 2.5. Now, we define the nonlinear function $\mathbf{g}(\mathbf{x})$ and propagate the sigma points generated above by using the weight $\pi_0 = 0.75$. The nonlinear function we use is given as

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} -\sqrt{|x^2 + 2y|} \\ \sqrt{x^2 + y^2} \end{bmatrix}. \tag{2.51}$$

The propagated particles and sigma points with their $\%99$ confidence ellipses are shown in Figure 2.6. Note that the mean and covariance obtained using sigma

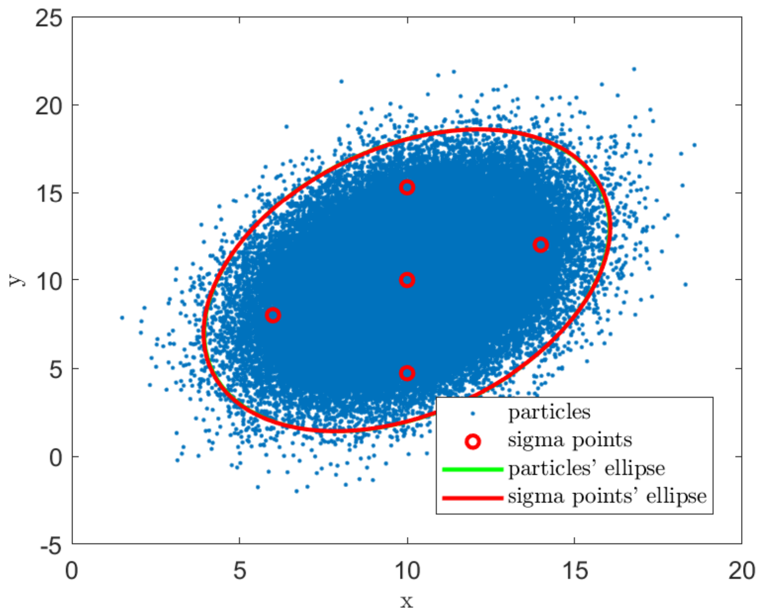



Figure 2.5: Particles and sigma points generated from the distribution given in equation (2.50). $\%99$ confidence ellipses are formed by using particles and sigma points. The weight $\pi_0$ is taken as $0.5$.

points are similar to those calculated using particles.

### 2.3 Kullback-Leibler Divergence

The aim of divergence metrics is to measure the similarity/dissimilarity between probability distributions. When two probability distributions are similar, the divergence becomes low. There are different divergence measures defined in the literature. In this thesis, we are interested in Kullback-Leibler divergence. Since the KL divergence is not symmetric, there are two versions of KL divergence, namely forward and reverse KL divergence. Complicated probability distributions can be approximated with a proposed distribution by minimizing divergence metrics between them. We assume that a true distribution $p(\mathbf{x})$ is desired to be approximated with a proposed distribution $q(\mathbf{x})$. To find the distribution $q(\mathbf{x})$, we define divergence metrics then we minimize them. We first define the forward KL divergence between the distributions $p(\mathbf{x})$ and

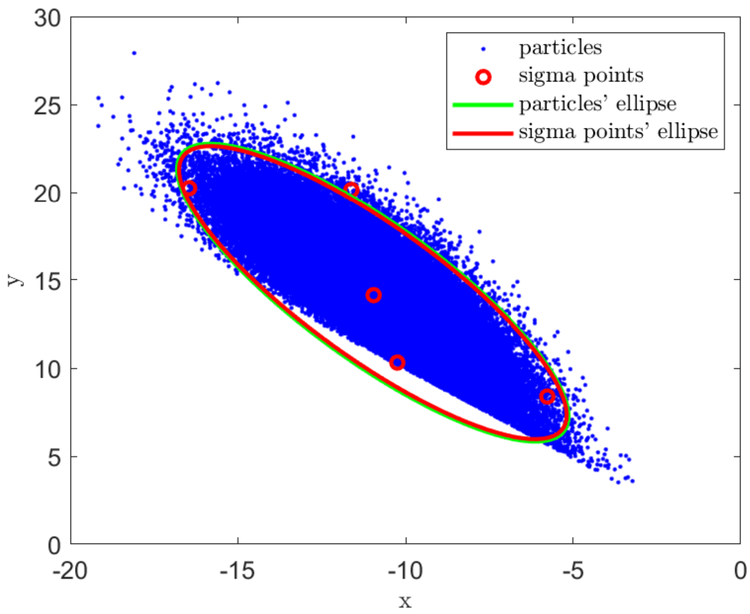



Figure 2.6: Particles and sigma points propagated through $\mathbf{g}(\mathbf{x})$ given in (2.51). $\%99$ confidence ellipses are formed by using particles and sigma points.

$q(\mathbf{x})$ as

$$\mathrm{KL}(q(\mathbf{x})||p(\mathbf{x})) = \int q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x})} \, \mathrm{d}\mathbf{x}. \tag{2.52}$$

Note that the KL divergence is nonnegative and equals zero only when the two PDFs are identical. While minimizing this divergence measure, if $p(\mathbf{x})$ is small in some regions, $q(\mathbf{x})$ also becomes small in that regions. Note that when $p(\mathbf{x})$ is zero, $q(\mathbf{x})$ also needs to be zero. Due to this property, this divergence measure is called zero forcing [2].

The reverse KL divergence for the distributions $p(\mathbf{x})$ and $q(\mathbf{x})$ is given below:

$$\mathrm{KL}(p(\mathbf{x})||q(\mathbf{x})) = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} \, \mathrm{d}\mathbf{x}. \tag{2.53}$$

As it can be seen from the equations (2.52) and (2.53), the KL divergence is not symmetric. This property prevents it from being a valid distance measure. While minimizing the reverse KL divergence, $q(\mathbf{x})$ tries to cover the whole support of $p(\mathbf{x})$. When $p(\mathbf{x})$ is greater than zero, $q(\mathbf{x})$ needs to be greater than zero. Due to this behavior, the reverse KL divergence is an inclusive divergence measure [2].

Minimization of the reverse KL divergence approximates the original probability density function (PDF) by covering the whole support where the PDF exists. In some scenarios, this can cause a problem. For example, assume that we try to approximate a bimodal Gaussian mixture distribution $p(\mathbf{x})$ with a normal distribution $q(\mathbf{x})$. In addition, suppose that one of the modes of $p(\mathbf{x})$ has a small probability and the modes are separated far from each other. Minimizing the reverse KL divergence results in a normal distribution $q(\mathbf{x})$ with a large variance in order to cover the mode having low probability. In this situation, using forward KL divergence may be more useful because of the zero forcing property.

Since $\alpha$-divergence covers both forward and reverse KL divergences, we also mention it. The formula for the $\alpha$-divergence is given as

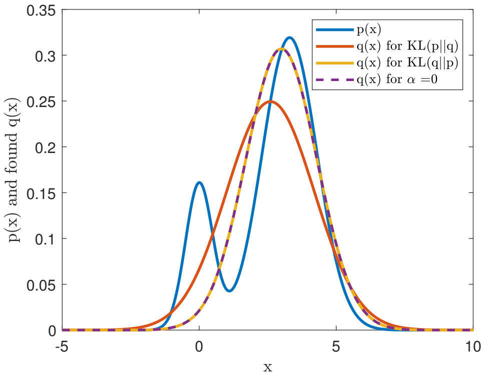
$$D_\alpha(p(\mathbf{x})||q(\mathbf{x})) = \frac{1}{\alpha(1-\alpha)}\left(1 - \int p^\alpha(\mathbf{x})q^{1-\alpha}(\mathbf{x})\,\mathrm{d}\mathbf{x}\right). \tag{2.54}$$

When $\alpha \to 0$ and $\alpha \to 1$, we obtain forward KL divergence and reverse KL divergence, respectively. Another special case is $\alpha = 0.5$. In this case, the divergence measure is called Hellinger distance which is a valid distance metric [2,3]. To demonstrate the relationship between these three divergence measures, we look at a simple scenario in which we try to approximate a bimodal Gaussian mixture distribution with a normal distribution. The distribution that we want to approximate is given below:

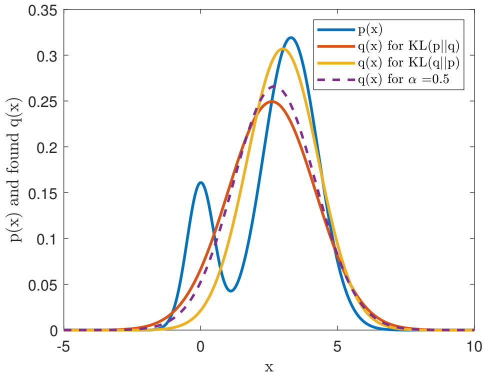
$$p(\mathbf{x}) = \pi_1 p_1(\mathbf{x}) + \pi_2 p_2(\mathbf{x}), \tag{2.55}$$

where $\pi_1 = 0.2$, $\pi_2 = 0.8$ and both $p_1(\mathbf{x})$ and $p_2(\mathbf{x})$ are Gaussian PDFs. The mean and standard deviation of $p_1(\mathbf{x})$ are $0$ and $0.5$, respectively. The mean and standard deviation of $p_2(\mathbf{x})$ are $3.3$ and $1$, respectively. As a result of minimizing related divergence measures, we obtain approximate Gaussian distributions $q(\mathbf{x})$ shown in Figure 2.7.

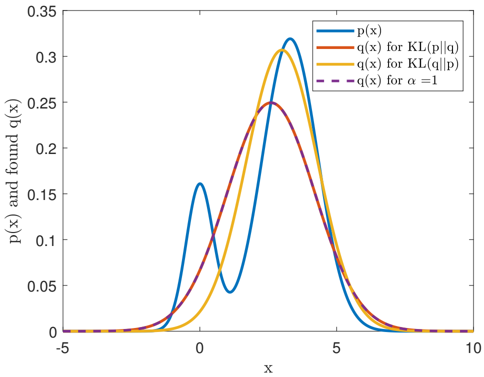
When the Figure 2.7 is examined, it is realized that forward KL divergence and $\alpha$ divergence for $\alpha = 0$ case give the same result. Similar to that result, when $\alpha = 1$, the reverse KL divergence and $\alpha$ divergence provide same results. In addition, when the value of $\alpha$ increases, the variance of the proposed distribution increases. Similarly, the variance of $q(\mathbf{x})$ determined by reverse KL divergence minimization is greater than the variance of $q(\mathbf{x})$ determined by forward KL divergence minimization.




(a) $\alpha \to 0$




(b) $\alpha = 0.5$




(c) $\alpha \to 1$

Figure 2.7: The distribution $p(\mathbf{x})$ and its approximations found via minimization of forward KL divergence, reverse KL divergence and $\alpha$-divergence.

## 2.4 Newton's Method

Newton-Raphson method is an iterative tool that is used to find the roots of a function. Assume that we have a function $g(x)$ and one of its root is $a$ such that $g(a) = 0$. To obtain the value of $a$, we start with an initial estimate $x_0$. Then, our estimate is improved to $x_1$ by using $x_0$. We continue improving our estimates until reaching the root $a$. Now, we let our estimate $x_0$ to be $a = x_0 + h$. We assume that $h$ is sufficiently small such that the following linear approximation can be performed

$$g(a) = g(x_0 + h) \approx g(x_0) + hg'(x_0). \tag{2.56}$$

The value of $h$ can be obtained by equating the above equation to $0$.

$$h \approx -\frac{g(x_0)}{g'(x_0)}. \tag{2.57}$$

The value of $a$ can be written by using $h$ above as

$$a = x_0 + h \approx x_0 - \frac{g(x_0)}{g'(x_0)}. \tag{2.58}$$

We can think the last expression as the improved estimate of $a$ and we call it $x_1$. Then, $x_1$ can be written as

$$x_1 = x_0 - \frac{g(x_0)}{g'(x_0)}. \tag{2.59}$$

At the next step, $x_2$ is obtained from $x_1$ similar to above equation.

$$x_2 = x_1 - \frac{g(x_1)}{g'(x_1)}. \tag{2.60}$$

We continue this procedure until reaching the value $a$. So, the general expression for Newton-Raphson method is given as

$$x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)}. \tag{2.61}$$

It is worth noting that when we approach the value of $a$, $g(x_n)$ approaches $0$ and the difference between $x_{n+1}$ and $x_n$ gets extremely small. At convergence, $g(x_n)$ becomes $0$ and we obtain the value of $x_n$ as $a$. After this point, the value of $x_n$ which is equal to $a$ does not change.

We present a geometric explanation of the Newton-Raphson technique in Figure 2.8. We first start with the initial point $x_0$. The function $g(\cdot)$ is approximated with a tangent line at point $x_0$. The slope of the tangent line at point $x_0$ is written as

$$g'(x_0) = \frac{g(x_0)}{x_0 - x_1}. \tag{2.62}$$

When we solve for $x_1$, we obtain the following.

$$x_1 = x_0 - \frac{g(x_0)}{g'(x_0)}. \tag{2.63}$$

By using the same procedure, $x_2$ is calculated from $x_1$. We continue this iteration until reaching convergence. As stated previously, we want to minimize a cost function in the nonlinear filtering problem in Chapter 3 of this thesis. To perform this, we take

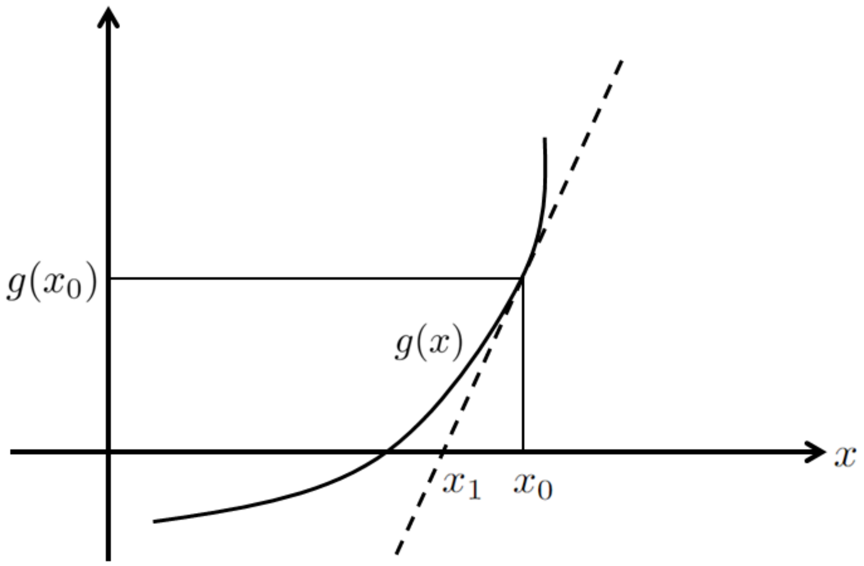



Figure 2.8: Geometrical representation of Newton-Raphson method.

the derivative of the cost function and equate it to zero. Hence, we try to find the roots of the derivative of the cost function. However, this problem cannot be solved analytically and Newton's method is used. Assume that we try to minimize a function $g(x)$. To achieve this, we need to find the roots of function $g'(x)$. The roots are obtained by using Newton's method given in (2.61) as follows:

$$x_{n+1} = x_n - \frac{g'(x_n)}{g''(x_n)}, \tag{2.64}$$

where $g''(x)$ is the second derivative of the function $g(x)$. As can be realized, the function that we investigate is one-dimensional. When the dimension is greater than one, we can write the Newton update in (2.64) as

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \left(\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^{\mathrm{T}} g(\mathbf{x}_n)\right)^{-1} \nabla_{\mathbf{x}} g(\mathbf{x}_n), \tag{2.65}$$

where $\nabla_{\mathbf{x}} g(\mathbf{x}_n)$ is the gradient of $g(\mathbf{x})$ evaluated at $\mathbf{x}_n$ and $\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^{\mathrm{T}} g(\mathbf{x}_n)$ is the Hessian matrix of $g(\mathbf{x})$ at $\mathbf{x}_n$. Newton's method can be modified by including a constant step size $\alpha$ whose value is between $0$ and $1$ as follows [4]:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \alpha \left(\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^{\mathrm{T}} g(\mathbf{x}_n)\right)^{-1} \nabla_{\mathbf{x}} g(\mathbf{x}_n). \tag{2.66}$$

The inclusion of the step size allows for faster convergence. The value of $\alpha$ can be changed by using some conditions such as the Wolfe conditions and the Goldstein conditions [5]. These conditions prevent divergence while also allowing for fast convergence.

## 2.5 Inverse Wishart Distribution

Inverse Wishart distribution is the generalization of the inverse gamma distribution for the positive definite matrices. There are different expressions for the IW distribution in the literature. We use the following expression given in [6].

$$\mathcal{IW}(\mathbf{X}; v, \mathbf{V}) = \frac{|\mathbf{V}|^{\frac{1}{2}(v-d-1)} \operatorname{etr}\left(-\frac{1}{2}\mathbf{X}^{-1}\mathbf{V}\right)}{2^{\frac{1}{2}(v-d-1)d}\Gamma_d\left[\frac{1}{2}(v-d-1)\right]|\mathbf{X}|^{\frac{v}{2}}}, \quad \mathbf{X} > \mathbf{0}, \mathbf{V} > \mathbf{0}, v > 2d, \tag{2.67}$$

where $\mathbf{X}$ is a $d \times d$ positive definite random matrix. The parameters $v$ and $\mathbf{V}$ denote degrees of freedom and scale matrix, respectively. $\mathbf{X} > \mathbf{0}$ indicates that the matrix is positive definite. $\operatorname{etr}$ is the exponential trace function and $\Gamma_d$ is a multivariate Gamma function. Lastly, the operator $|\cdot|$ represents the determinant. The mean of the random matrix $\mathbf{X}$ is given as

$$\mathrm{E}[\mathbf{X}] = \frac{\mathbf{V}}{v - 2d - 2}, \quad v > 2d + 2. \tag{2.68}$$

The cross covariance between matrix elements $\mathbf{X}_{ij}$ and $\mathbf{X}_{kl}$ is given as

$$\operatorname{Cov}(\mathbf{X}_{ij}, \mathbf{X}_{kl}) = \frac{2(v-2d-2)^{-1}\mathbf{V}_{ij}\mathbf{V}_{kl} + \mathbf{V}_{ik}\mathbf{V}_{jl} + \mathbf{V}_{il}\mathbf{V}_{kj}}{(v-2d-1)(v-2d-2)(v-2d-4)}, \quad v > 2d + 4, \tag{2.69}$$

where $\mathbf{V}_{ij}$ is the element of matrix $\mathbf{V}$ at $i^{th}$ row and $j^{th}$ column. The variance of the diagonal elements $\mathbf{X}_{ii}$ is given as

$$\operatorname{Var}(\mathbf{X}_{ii}) = \frac{2\mathbf{V}_{ii}^2\left((v-2d-2)^{-1} + 1\right)}{(v-2d-1)(v-2d-2)(v-2d-4)}, \tag{2.70}$$

for $i = 1, ..., d$.

IW distribution is the conjugate prior of the covariance parameter for multivariate Gaussian distributed data. We assume that we have $n$ observations $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n]$ where $\mathbf{y}_i \sim \mathcal{N}(\mathbf{y}_i; \mathbf{0}, \mathbf{X})$. The measurements $\mathbf{y}_i$ are conditionally independent given $\mathbf{X}$. We assume that the covariance matrix is random and it has inverse Wishart distribution, i.e., $p(\mathbf{X}) = \mathcal{IW}(\mathbf{X}; v, \mathbf{V})$. To find the posterior distribution of $\mathbf{X}$, we use Bayes' rule.

$$p(\mathbf{X}|\mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X})p(\mathbf{X})}{p(\mathbf{Y})} \tag{2.71a}$$

$$\propto p(\mathbf{Y}|\mathbf{X})p(\mathbf{X}) \tag{2.71b}$$

$$\propto \left(\prod_{i=1}^{n} |\mathbf{X}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\mathbf{y}_i^{\mathrm{T}}\mathbf{X}^{-1}\mathbf{y}_i\right)\right) \exp\left(-\frac{1}{2}\operatorname{tr}(\mathbf{X}^{-1}\mathbf{V})\right) |\mathbf{X}|^{-\frac{v}{2}} \tag{2.71c}$$

$$= |\mathbf{X}|^{-\frac{v+n}{2}} \exp\left(-\frac{1}{2}\operatorname{tr}\left(\mathbf{X}^{-1}\sum_{i=1}^{n}\mathbf{y}_i\mathbf{y}_i^{\mathrm{T}}\right)\right) \exp\left(-\frac{1}{2}\operatorname{tr}(\mathbf{X}^{-1}\mathbf{V})\right) \tag{2.71d}$$

$$= |\mathbf{X}|^{-\frac{v+n}{2}} \exp\left(-\frac{1}{2}\operatorname{tr}\left(\mathbf{X}^{-1}(\mathbf{V}+n\mathbf{C})\right)\right) \tag{2.71e}$$

$$\propto \mathcal{IW}(\mathbf{X}; v+n, \mathbf{V}+n\mathbf{C}), \tag{2.71f}$$

where $\mathbf{C}$ is the sample covariance matrix, which is denoted as

$$\mathbf{C} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{y}_i\mathbf{y}_i^{\mathrm{T}}. \tag{2.72}$$

Hence, the prior distribution of $\mathbf{X}$ is IW and the posterior distribution is also IW. As a result, the IW distribution is the conjugate prior for the covariance of the multivariate Gaussian distribution. Finally, we investigate the mean of the posterior distribution of $\mathbf{X}$.

$$\mathrm{E}\left[\mathbf{X}|\mathbf{Y}\right] = \frac{\mathbf{V}+n\mathbf{C}}{v+n-2d-2} \tag{2.73a}$$

$$= \frac{n}{v+n-2d-2}\mathbf{C} + \left(1 - \frac{n}{v+n-2d-2}\right)\frac{\mathbf{V}}{v-2d-2} \tag{2.73b}$$

$$= \frac{n}{v+n-2d-2}\mathbf{C} + \left(1 - \frac{n}{v+n-2d-2}\right)\mathrm{E}[\mathbf{X}] \tag{2.73c}$$

As can be seen, the posterior mean is a weighted average of prior mean $\mathrm{E}[\mathbf{X}]$ and the sample covariance matrix $\mathbf{C}$. Furthermore, if we let the number of observations $n$ go to infinity, the posterior mean equals the sample covariance matrix $\mathbf{C}$.

# CHAPTER 3

# GAUSSIAN MIXTURE FILTERING WITH NONLINEAR MEASUREMENTS MINIMIZING FORWARD KULLBACK-LEIBLER DIVERGENCE

In this chapter, a Gaussian mixture filter is proposed for the state estimation of dynamical systems with nonlinear measurements. The filter is derived by solving an assumed density filtering problem where the Kullback-Leibler divergence from the assumed posterior, which is a Gaussian mixture, to the true posterior (which we call forward KL divergence) is minimized. The approximate solution to this problem gives an iterative measurement update by which the weights, means and covariances of the assumed posterior are optimized. The resulting Gaussian mixture filter is shown to be a generalization of the (damped) posterior linearization filter to Gaussian mixture posteriors. The performance of the proposed filter is illustrated and compared to alternatives on a challenging example of target tracking in a sensor network. The results show that the proposed filter can outperform Gaussian filters as well as the Gaussian sum filter yielding results very close to a bootstrap particle filter when the number of components in the assumed posterior is sufficiently large.

## 3.1 Introduction

Kalman filter is widely utilized for the state estimation of dynamical systems in many different applications like audio signal processing [7], target tracking [8], navigation [9] and finance [10]. As a Bayesian estimator, Kalman filter estimates the posterior density of the state as a Gaussian distribution whose mean is minimum mean square error (MMSE) state estimate for linear Gaussian systems and linear MMSE state estimate for linear non-Gaussian systems. When the dynamical system of interest is nonlinear, calculating the exact posterior is almost always analytically and computationally intractable and hence one must resort to approximations.

The most common solution for the nonlinear state estimation problem is to linearize the system of interest as in extended Kalman filter (EKF) [11], iterated extended Kalman filter (IEKF) [11], statistically linearized filter (SLF) [11, 12] and (damped) posterior linearization filter (PLF) [13, 14]. The second type of approaches is the so-called sigma-point approaches, where the expectations of nonlinear functions (i.e., Gaussian integrals) appearing in nonlinear state estimation are approximated by weighted sums of the values of the corresponding functions at a finite set of deterministically selected points in the state space, called sigma-points. Unscented Kalman filter (UKF) [1], Gauss-Hermite Kalman filter (GHKF) [15] and cubature Kalman filter (CKF) [16] are the common examples of the sigma-point approaches. All of the aforementioned filters above result in approximate Gaussian posteriors. The third approach to nonlinear state estimation approximates the intractable posterior with a Gaussian mixture as in the Gaussian sum filter (GSF) [17, 18] where multiple Gaussian filters (e.g., EKF, UKF) are run in parallel to obtain a better representation of the posterior. The fourth and final type of approaches to deal with nonlinear state estimation is the grid based approaches in which the intractable posterior is approximated with a Dirac delta mixture. The components of the mixture are located either on a fixed deterministic grid of points in the state-space as in the point mass filter (PMF) [19–21] or on an adaptive stochastic grid of points as in the particle filter (PF) [22]. Particle filters belong to a larger class of algorithms called sequential Monte Carlo methods [23] and there is a vast related literature about them (See the tutorials [24–28] and the references therein).

In this thesis, we propose a Gaussian mixture filter for the state estimation of dynamical systems with nonlinear measurements. In order to derive the filter, we solve an assumed density filtering problem approximately where we minimize the Kullback-Leibler (KL) divergence from the assumed posterior, which is a Gaussian mixture, to the true posterior (which we call *forward KL divergence*). The approximate solution to this problem gives an iterative measurement update by which the weights, means and covariances of the assumed posterior are optimized. The resulting Gaussian mixture filter is shown to be a generalization of the (damped) PLF to Gaussian mixture posteriors. The performance of the proposed filter is illustrated and compared to alternatives on a challenging example of target tracking in a sensor network [13]. The results show that the proposed filter can outperform Gaussian filters as well as GSF yielding results very close to a bootstrap particle filter when the number of components in the assumed posterior is sufficiently large.

The remainder of the chapter is organized as follows. Section 3.2 places the current work in the context of existing literature after presenting a brief survey of the related work. In Section 3.3, the problem definition is given. The main contribution of the chapter, the proposed iterative measurement update, is presented in Section 3.4. We illustrate the characteristics of the proposed measurement update on a static Bayesian estimation problem in Section 3.5. In Section 3.6, the performance of the proposed Gaussian mixture filter, called forward Kullback-Leibler filter (FKLF), is compared with alternatives on a challenging target tracking scenario. Conclusions are presented in Section 3.7. Finally, the derivations are provided in Section 3.8.

## 3.2 Related Literature

The literature on improving the performance of GSF can be roughly divided into two types of approaches. The first type of methods aims to decrease the Gaussian filter errors appearing in the updates of the mixture components in GSF. This is done by splitting the Gaussian components that would result in large Gaussian filter errors (e.g., linearization errors) into multiple Gaussian components with smaller covariances. Since a Gaussian component with a smaller covariance has a localized support in the state space, its Gaussian filter update would be less susceptible to approximation errors. Hence splitting components is expected to improve the performance of GSF at the expense of computation cost. The examples of splitting approaches can be found in the studies [29–35]. The second type of approaches tries to improve the update of the mixture weights in the time update [36] and measurement update [15] of GSF. In all of the aforementioned work, the mean and covariance of components in GSF are updated non-iteratively by Gaussian filters and the related approximations are based on the prior mixture parameters. The mean and covariance updates of the proposed filter in this work, on the other hand, are iterative and the update approximations are based on the posterior mixture parameters. Hence the proposed Gaussian mixture filter can be thought of the posterior version of GSF in the sense that PLF is the posterior version of SLF.

KL divergence minimization is quite common in Bayesian estimation. Two well known examples in this respect are variational methods [37] and expectation propagation (EP) [38, 39]. Variational methods make approximate Bayesian estimation by minimizing the KL divergence from the assumed posterior density to true posterior, i.e., by minimizing the forward KL divergence. EP, on the other hand, calculates approximate posteriors by minimizing the KL divergence from the true posterior to assumed posterior, i.e., by minimizing the reverse KL divergence.

There have been many attempts to apply KL divergence minimization to nonlinear filtering problems in the past. Forward KL divergence is minimized in [40] to obtain nonlinear filters which utilize Monte Carlo techniques to calculate the intractable expectations. In [41], an iterative nonlinear filter, similar to PLF, is proposed by minimizing forward KL divergence. Reverse KL divergence is minimized in [42] to obtain a nonlinear filter which utilizes Gauss-Hermite quadrature to calculate the intractable expectations. In [43] forward and reverse KL divergences and $\alpha$-divergence are minimized and intractable expectations are calculated using Monte Carlo methods to obtain nonlinear filters. Another nonlinear filter based on $\alpha$-divergence minimization with Monte Carlo integration is given in [44]. An analysis of Gaussian filter approximations in terms of reverse KL divergences between true and approximate joint distributions of state and the measurement is given in [45], which is later utilized in designing a new nonlinear filter in [46].

In all of the aforementioned work related to KL divergence minimization, the assumed posterior is a Gaussian distribution. In this respect, the current work, which applies KL minimization with a Gaussian mixture assumed posterior, fills the gap in the nonlinear filtering literature between Gaussian sum filtering and KL divergence minimization.

### 3.3 Problem Definition

We consider the Bayes update of the random vector $\mathbf{x} \in \mathbb{R}^{n_x}$ having the Gaussian mixture prior distribution $p(\mathbf{x})$ given as

$$p(\mathbf{x}) = \sum_{n=1}^{N} \pi_{0n} \underbrace{\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{0n}, \boldsymbol{\Sigma}_{0n})}_{\triangleq p_{0n}(\mathbf{x})}, \tag{3.1}$$

with the noisy measurements $\mathbf{y} \in \mathbb{R}^{n_y}$ given as

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) + \mathbf{v}, \tag{3.2}$$

where $\mathbf{h}(\cdot)$ is, in general, a nonlinear function and $\mathbf{v} \sim \mathcal{N}(\mathbf{v}; \mathbf{0}, \mathbf{R})$ represents the Gaussian measurement noise independent from $\mathbf{x}$. The joint distribution $p(\mathbf{y}, \mathbf{x})$ is given as

$$p(\mathbf{y}, \mathbf{x}) = \mathcal{N}(\mathbf{y}; \mathbf{h}(\mathbf{x}), \mathbf{R}) \sum_{n=1}^{N} \pi_{0n} \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{0n}, \boldsymbol{\Sigma}_{0n}). \tag{3.3}$$

The exact posterior $p(\mathbf{x}|\mathbf{y})$ is, most of the times, analytically intractable. Therefore it is of interest to come up with approximations for the posterior. In assumed density filtering, the posterior is assumed to have a known parametric form whose parameters are estimated using an optimization problem.

In this work we assume that the approximate posterior $q(\mathbf{x})$ is a Gaussian mixture, i.e.,

$$q(\mathbf{x}) = \sum_{j=1}^{J} \pi_j \underbrace{\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_j, \boldsymbol{\Sigma}_j)}_{\triangleq q_j(\mathbf{x})}. \tag{3.4}$$

We would like to find the parameters $\pi_{1:J}$,$\hat{\mathbf{x}}_{1:J}$,$\boldsymbol{\Sigma}_{1:J}$ of the assumed posterior by minimizing the forward KL divergence between $q(\cdot)$ and the true posterior $p(\mathbf{x}|\mathbf{y})$.[1] The KL divergence we are interested in can be written as

$$\mathrm{KL}(q(\cdot)||p(\cdot|\mathbf{y})) \triangleq \int q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x}|\mathbf{y})} \,\mathrm{d}\mathbf{x} \tag{3.5a}$$

$$\stackrel{+}{=} \int q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{y}, \mathbf{x})} \,\mathrm{d}\mathbf{x} \tag{3.5b}$$

$$= \int q(\mathbf{x}) \log q(\mathbf{x}) \,\mathrm{d}\mathbf{x} - \int q(\mathbf{x}) \log p(\mathbf{x}) \,\mathrm{d}\mathbf{x} - \int q(\mathbf{x}) \log p(\mathbf{y}|\mathbf{x}) \,\mathrm{d}\mathbf{x} \tag{3.5c}$$

[1] The notation $x_{a:b}$ denotes $\{x_j\}_{j=a}^{b}$ for any two integers $a$ and $b$ satisfying $b \geq a$.

$$= \mathrm{E}_q[\log q(\mathbf{x})] - \mathrm{E}_q[\log p(\mathbf{x})] - \mathrm{E}_q[\log p(\mathbf{y}|\mathbf{x})], \tag{3.5d}$$

where the sign $\stackrel{+}{=}$ denotes equality up to an additive constant with respect to the variables $\pi_{1:J}$,$\hat{\mathbf{x}}_{1:J}$,$\mathbf{\Sigma}_{1:J}$; the subscript of an expectation operator denotes the density with respect to which the expectation has to be taken. As a result, we get

$$\mathrm{KL}(q(\cdot)||p(\cdot|\mathbf{y})) \stackrel{+}{=} \mathrm{E}_q[\log q(\mathbf{x})] - \mathrm{E}_q[\log p(\mathbf{x})] - \mathrm{E}_q[\log p(\mathbf{y}|\mathbf{x})]. \tag{3.6}$$

In this work we would like to find a possibly approximate solution to the following optimization problem.

$$\pi^*_{1:J}, \hat{\mathbf{x}}^*_{1:J}, \mathbf{\Sigma}^*_{1:J} = \arg \min_{\pi_{1:J},\hat{\mathbf{x}}_{1:J},\mathbf{\Sigma}_{1:J}} \mathrm{KL}(q(\cdot)||p(\cdot|\mathbf{y})), \tag{3.7}$$

which is going to yield the approximate posterior $q^*(\mathbf{x}) \triangleq \sum_{j=1}^{J} \pi^*_j \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}^*_j, \mathbf{\Sigma}^*_j)$. The difficulties associated with the optimization problem (3.7) can be listed as follows.

- The expectations on the right hand side of (3.6) are intractable and hence the exact cost cannot be calculated analytically;
- Even when the expectations on the right hand side of (3.6) can be taken, it is not possible to obtain analytical (argument of the) minimum for the resulting cost function;
- The optimization problem is non-convex and it has many local minima.

In order to overcome these problems, we are going to

- make approximations to calculate the intractable expectations;
- use iterative methods to handle the optimization;
- search only for a local minimum instead of the global minimum.

### 3.4 Proposed Iterative Measurement Update

For the sake of brevity we first call the right hand side of (3.6) as the cost function $L(q)$, i.e.,

$$L(q) \triangleq \mathrm{E}_q[\log q(\mathbf{x})] - \mathrm{E}_q[\log p(\mathbf{x})] - \mathrm{E}_q[\log p(\mathbf{y}|\mathbf{x})]. \tag{3.8}$$

In the following subsections, we will first obtain implicit expressions for the means $\hat{\mathbf{x}}_{1:J}$, the covariances $\mathbf{\Sigma}_{1:J}$ and the weights $\pi_{1:J}$ and then use those implicit expressions to form an iterative procedure to minimize $L(q)$.

### 3.4.1 Implicit Expressions for the Means and Covariances

The gradients of the cost $L(q)$ with respect to the mean $\hat{\mathbf{x}}_i$ and covariance $\mathbf{\Sigma}_i$ are approximately given as (See Section 3.8.1 for derivations)

$$\nabla_{\hat{\mathbf{x}}_i} L(q) \approx -\pi_i \sum_{j=1}^{J} \gamma_{i,j} \mathbf{\Sigma}_j^{-1} (\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j)$$

$$+ \pi_i \sum_{n=1}^{N} \beta_{i,n} \mathbf{\Sigma}_{0n}^{-1} (\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_{0n}) - \pi_i \, \mathrm{E}_{q_i} \left[ \nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x}) \right], \tag{3.9a}$$

$$\nabla_{\mathbf{\Sigma}_i} L(q) \approx -\frac{1}{2} \pi_i \sum_{j=1}^{J} \gamma_{i,j} \mathbf{\Sigma}_j^{-1} + \frac{1}{2} \pi_i \sum_{n=1}^{N} \beta_{i,n} \mathbf{\Sigma}_{0n}^{-1}$$

$$- \frac{1}{2} \pi_i \, \mathrm{E}_{q_i} \left[ \nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^{\mathrm{T}} \log p(\mathbf{y}|\mathbf{x}) \right], \tag{3.9b}$$

where $\gamma_{i,j}$ and $\beta_{i,n}$ are responsibilities defined as

$$\gamma_{i,j} \triangleq \frac{\pi_j \mathcal{N}(\hat{\mathbf{x}}_i; \hat{\mathbf{x}}_j, \mathbf{\Sigma}_j)}{\sum_{\ell=1}^{J} \pi_\ell \mathcal{N}(\hat{\mathbf{x}}_i; \hat{\mathbf{x}}_\ell, \mathbf{\Sigma}_\ell)}, \tag{3.10a}$$

$$\beta_{i,n} \triangleq \frac{\pi_{0n} \mathcal{N}(\hat{\mathbf{x}}_i; \hat{\mathbf{x}}_{0n}, \mathbf{\Sigma}_{0n})}{\sum_{\ell=1}^{N} \pi_{0\ell} \mathcal{N}(\hat{\mathbf{x}}_i; \hat{\mathbf{x}}_{0\ell}, \mathbf{\Sigma}_{0\ell})}, \tag{3.10b}$$

for $i, j \in \{1, \ldots, J\}$ and $n \in \{1, \ldots, N\}$. The symbol $\nabla_{\mathbf{D}}$ denotes the gradient operator with respect to $\mathbf{D} \in \mathbb{R}^{n \times m}$, whose elements are described by

$$[\nabla_{\mathbf{D}}]_{i,j} \triangleq \frac{\partial}{\partial [\mathbf{D}]_{i,j}}, \tag{3.11}$$

for $1 \leq i \leq n$, $1 \leq j \leq m$, where the notation $[\cdot]_{i,j}$ denotes the $i,j$th element of the argument matrix.

**Remark 1.** *When $N = 1$ and $J = 1$, i.e., when we have the Gaussian case with $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_0, \mathbf{\Sigma}_0)$ and $q(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{\Sigma})$, the approximations in* (3.9) *becomes exact giving the following expressions.*

$$\nabla_{\hat{\mathbf{x}}} L(q) = \mathbf{\Sigma}_0^{-1} (\hat{\mathbf{x}} - \hat{\mathbf{x}}_0) - \mathrm{E}_q \left[ \nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x}) \right], \tag{3.12a}$$

$$\nabla_{\mathbf{\Sigma}} L(q) = -\frac{1}{2} \mathbf{\Sigma}^{-1} + \frac{1}{2} \mathbf{\Sigma}_0^{-1} - \frac{1}{2} \mathrm{E}_q \left[ \nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^{\mathrm{T}} \log p(\mathbf{y}|\mathbf{x}) \right]. \tag{3.12b}$$

*Moreover, when the measurements* $\mathbf{y}$ *are linear, i.e., the function* $\mathbf{h}(\mathbf{x})$ *satisfies* $\mathbf{h}(\mathbf{x}) = \mathbf{H}\mathbf{x}$*, the expressions in* (3.12) *turn into the following.*

$$\nabla_{\hat{\mathbf{x}}} L(q) \approx \mathbf{\Sigma}_0^{-1}(\hat{\mathbf{x}} - \hat{\mathbf{x}}_0) - \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}), \tag{3.13a}$$

$$\nabla_{\mathbf{\Sigma}} L(q) \approx -\frac{1}{2}\mathbf{\Sigma}^{-1} + \frac{1}{2}\mathbf{\Sigma}_0^{-1} + \frac{1}{2}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}. \tag{3.13b}$$

*When we equate the gradients in* (3.13) *to zero and solve for* $\hat{\mathbf{x}}$ *and* $\mathbf{\Sigma}$*, we get the Kalman filter update equations.*

**Remark 2.** *Let* $J = 1$*. Suppose that we find a solution* $\hat{\mathbf{x}}^*, \mathbf{\Sigma}^*$ *to the equations*

$$\nabla_{\hat{\mathbf{x}}} L(q) = \mathbf{0}, \qquad \nabla_{\mathbf{\Sigma}} L(q) = \mathbf{0}, \tag{3.14}$$

*giving us a local optimum of the optimization problem* (3.7)*. Now let* $J = 2$*. Then the selection*

$$\hat{\mathbf{x}}_j^* = \hat{\mathbf{x}}^*, \qquad \mathbf{\Sigma}_j^* = \mathbf{\Sigma}^*, \tag{3.15}$$

*for* $j = 1, 2$*, satisfies the equations*

$$\nabla_{\hat{\mathbf{x}}_i} L(q) = \mathbf{0}, \qquad \nabla_{\mathbf{\Sigma}_i} L(q) = \mathbf{0}, \tag{3.16}$$

*for* $i = 1, 2$*, for any selection of the weights* $\pi_{1:2}$*. Hence a local optimum of* (3.7) *for* $J = 1$ *with* $q^*(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}^*, \mathbf{\Sigma}^*)$ *is a local optimum of* (3.7) *for* $J = 2$ *with* $q^*(\mathbf{x}) = \sum_{j=1}^{2} \pi_j \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_j^*, \mathbf{\Sigma}_j^*)$ *where the parameters* $\hat{\mathbf{x}}_{1:2}^*, \mathbf{\Sigma}_{1:2}^*$ *are given as in* (3.15)*. Note that this is the general property of the local solutions of the optimization problem* (3.7)*, i.e., the solution for some* $J = J_1$ *is also a solution for* $J = J_2 > J_1$*. As a result, when one looks for only a local optimum of* (3.15)*, the Gaussian components in the solution might be identical, effectively reducing the number of components of the Gaussian mixture. When the* $i$*th and* $j$*th components in the solution are identical, the corresponding term of the first summation on the right hand side of the gradient* $\nabla_{\hat{\mathbf{x}}_i} L(q)$ *in* (3.9a)*, i.e.,* $\gamma_{i,j}\mathbf{\Sigma}_j^{-1}(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j)$ *vanishes. In the following, we will denote the indices of the Gaussian components which are identical to the* $i$*th component (in the local optimum) as* $\mathcal{N}_i$*, i.e.,*

$$\mathcal{N}_i \triangleq \{j \in \{1, \ldots, J\} | \hat{\mathbf{x}}_j = \hat{\mathbf{x}}_i, \mathbf{\Sigma}_j = \mathbf{\Sigma}_i\}. \tag{3.17}$$

**Remark 3.** *We are going to propose a Newton-type iteration for the means* $\hat{\mathbf{x}}_{1:J}$*. This means that we are going to propagate the means* $\hat{\mathbf{x}}_{1:J}$ *in a direction that makes* *an angle $\theta$ with the negative gradient satisfying $\cos(\theta) > 0$.*[2] *The negative gradient $-\nabla_{\hat{\mathbf{x}}_i} L(q)$ based on (3.9a) is composed of three terms with their corresponding effects on the $i$th mean $\hat{\mathbf{x}}_i$ described below:*

- *Term-1: $\pi_i \sum_{j=1}^{J} \gamma_{i,j} \mathbf{\Sigma}_j^{-1}(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j)$. This term pushes the $i$th mean $\hat{\mathbf{x}}_i$ away from the means of the non-identical components of the posterior mixture $q(\mathbf{x})$, i.e., $\hat{\mathbf{x}}_j$, $j \notin \mathcal{N}_i$, in a Newton-type iteration.*[3] *The responsibilities $\{\gamma_{i,j}\}_{j \notin \mathcal{N}_i}$ adjust the amount of push based on the closeness of the $i$th mean to the means of the other components, i.e., the push is stronger for close components to $\hat{\mathbf{x}}_i$. Note that since the non-identical components are pushed away from $i$th component, we expect the non-identical components to be far away from the $i$th component at the optimal solution. As a result the responsibility values of the identical components ($\gamma_{i,j}$, $j \in \mathcal{N}_i$) are expected to be much larger than those of the non-identical components ($\gamma_{i,j}$, $j \notin \mathcal{N}_i$).*
- *Term-2: $-\pi_i \sum_{n=1}^{N} \beta_{i,n} \mathbf{\Sigma}_{0n}^{-1}(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_{0n})$. This term pulls the $i$th mean $\hat{\mathbf{x}}_i$ towards the means $\hat{\mathbf{x}}_{0n}$, $n \in \{1, \ldots, N\}$, of the prior $p(\mathbf{x})$ in a Newton-type iteration. The responsibilities $\{\beta_{i,n}\}_{n=1}^{N}$ adjust the amount of pull based on the closeness of the $i$th mean to the means of the prior components, i.e., the pull is stronger for close components to $\hat{\mathbf{x}}_i$.*
- *Term-3: $\pi_i \, \mathrm{E}_{q_i} [\nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x})]$. This term pulls the $i$th mean $\hat{\mathbf{x}}_i$ towards the regions of the state space where the measurement $\mathbf{y}$ has a high likelihood.*

*Due to Term-1, we expect, at the optimal solution, that $\gamma_{i,j_1} \gg \gamma_{i,j_2}$ for $j_1 \in \mathcal{N}_i$ and $j_2 \notin \mathcal{N}_i$. Since we have $\sum_{j=1}^{J} \gamma_{i,j} = 1$ by definition, the approximations $\sum_{j \in \mathcal{N}_i} \gamma_{i,j} \approx 1$ and $\gamma_{i,j} \approx 0$, $j \notin \mathcal{N}_i$ are justified when necessary.*

Equating the gradient $\nabla_{\mathbf{\Sigma}_i} L(q)$ in (3.9b) to zero and assuming that $\pi_i \neq 0$, we get

$$\mathbf{0} = -\sum_{j=1}^{J} \gamma_{i,j} \mathbf{\Sigma}_j^{-1} + \sum_{n=1}^{N} \beta_{i,n} \mathbf{\Sigma}_{0n}^{-1} - \mathrm{E}_{q_i} \left[ \nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^{\mathrm{T}} \log p(\mathbf{y}|\mathbf{x}) \right] \tag{3.18a}$$

[2] This is due to the positive definiteness of the approximate Hessian we will use.

[3] This can be intuitively understood easily by considering a single quadratic $(\mathbf{x}-\mathbf{z})^{\mathrm{T}} \Sigma^{-1} (\mathbf{x}-\mathbf{z})$. Propagating an initial vector $\hat{\mathbf{x}}$ in the direction of the gradient $2\Sigma^{-1}(\hat{\mathbf{x}} - \mathbf{z})$ increases the value of the quadratic by moving $\hat{\mathbf{x}}$ away from $\mathbf{z}$ and vice versa. Note that Term-1 resembles the gradient of the quadratic sum $\pi_i \sum_{j=1}^{J} \gamma_{i,j} (\mathbf{x} - \hat{\mathbf{x}}_j)^{\mathrm{T}} \mathbf{\Sigma}_j^{-1} (\mathbf{x} - \hat{\mathbf{x}}_j)$. The related terms can be viewed as factors increasing the disagreement among the non-identical components of the posterior mixture as they are the reverse of the factors used in distributed estimation and control based on the idea of consensus [47].

$$= -\sum_{j\in\mathcal{N}_i}\gamma_{i,j}\boldsymbol{\Sigma}_j^{-1} - \sum_{j\notin\mathcal{N}_i}\gamma_{i,j}\boldsymbol{\Sigma}_j^{-1} + \sum_{n=1}^{N}\beta_{i,n}\boldsymbol{\Sigma}_{0n}^{-1}$$

$$- \mathrm{E}_{q_i}\left[\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathrm{T}}\log p(\mathbf{y}|\mathbf{x})\right] \tag{3.18b}$$

$$= -\boldsymbol{\Sigma}_i^{-1}\sum_{j\in\mathcal{N}_i}\gamma_{i,j} - \sum_{j\notin\mathcal{N}_i}\gamma_{i,j}\boldsymbol{\Sigma}_j^{-1} + \sum_{n=1}^{N}\beta_{i,n}\boldsymbol{\Sigma}_{0n}^{-1}$$

$$- \mathrm{E}_{q_i}\left[\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathrm{T}}\log p(\mathbf{y}|\mathbf{x})\right] \tag{3.18c}$$

$$\approx -\boldsymbol{\Sigma}_i^{-1} + \sum_{n=1}^{N}\beta_{i,n}\boldsymbol{\Sigma}_{0n}^{-1} - \mathrm{E}_{q_i}\left[\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathrm{T}}\log p(\mathbf{y}|\mathbf{x})\right], \tag{3.18d}$$

where we used the fact that $\boldsymbol{\Sigma}_j = \boldsymbol{\Sigma}_i$, $j \in \mathcal{N}_i$, and made the approximations $\sum_{j\in\mathcal{N}_i}\gamma_{i,j} \approx 1$ and $\gamma_{i,j} \approx 0$, $j \notin \mathcal{N}_i$. Solving now for $\boldsymbol{\Sigma}_i$, we can write the following implicit expression for the covariances.

$$\boldsymbol{\Sigma}_i \approx \left(\sum_{n=1}^{N}\beta_{i,n}\boldsymbol{\Sigma}_{0n}^{-1} - \mathrm{E}_{q_i}\left[\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathrm{T}}\log p(\mathbf{y}|\mathbf{x})\right]\right)^{-1}. \tag{3.19}$$

The implicit expression for the means $\{\hat{\mathbf{x}}_i\}_{i=1}^{J}$ we propose is given as follows.

$$\hat{\mathbf{x}}_i = \hat{\mathbf{x}}_i - \alpha\big(\nabla_{\hat{\mathbf{x}}_i}\nabla_{\hat{\mathbf{x}}_i}^{\mathrm{T}}L(q)\big)^{-1}\nabla_{\hat{\mathbf{x}}_i}L(q), \tag{3.20}$$

where $0 < \alpha \leq 1$ denotes the constant step-size. Note that the implicit expression in (3.20) would turn into a Newton-type update rule when used recursively. The Hessian $\nabla_{\hat{\mathbf{x}}_i}\nabla_{\hat{\mathbf{x}}_i}^{\mathrm{T}}L(q)$ is not positive definite in general, however, when we assume that the covariances $\boldsymbol{\Sigma}_{1:J}$ are close to their optimal values, i.e., when $\nabla_{\boldsymbol{\Sigma}_i}L(q) \approx \mathbf{0}$, $i = 1, \ldots, J$, the following approximation can be made (See Section 3.8.1 for a derivation).

$$\nabla_{\hat{\mathbf{x}}_i}\nabla_{\hat{\mathbf{x}}_i}^{\mathrm{T}}L(q) \approx \pi_i\boldsymbol{\Sigma}_i^{-1}. \tag{3.21}$$

Substitution of (3.9a) and (3.21) into (3.20) would give the following implicit expression for the means

$$\hat{\mathbf{x}}_i = \hat{\mathbf{x}}_i + \alpha\boldsymbol{\Sigma}_i\,\mathrm{E}_{q_i}[\nabla_{\mathbf{x}}\log p(\mathbf{y}|\mathbf{x})] + \alpha\boldsymbol{\Sigma}_i\sum_{j=1}^{J}\gamma_{i,j}\boldsymbol{\Sigma}_j^{-1}(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j)$$

$$- \alpha\boldsymbol{\Sigma}_i\sum_{n=1}^{N}\beta_{i,n}\boldsymbol{\Sigma}_{0n}^{-1}(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_{0n}), \tag{3.22}$$

for $i = 1, \ldots, J$.

### 3.4.2 Implicit Expression for the Weights

In order to optimize the weights $\pi_{1:J}$, we can perform constrained optimization using Lagrange multipliers technique with the cost function below.

$$\mathcal{L}(\pi_{1:J}) = L(q) + \lambda\left(1 - \sum_{j=1}^{J}\pi_j\right) \tag{3.23a}$$

$$= \mathrm{E}_q[\log q(\mathbf{x})] - \mathrm{E}_q[\log p(\mathbf{x})] - \mathrm{E}_q[\log p(\mathbf{y}|\mathbf{x})] + \lambda\left(1 - \sum_{j=1}^{J}\pi_j\right), \tag{3.23b}$$

where $\lambda \in \mathbb{R}$ is a Lagrange multiplier. We can write (3.23b) in terms of the weights $\pi_{1:J}$ approximately as follows (See Section 3.8.2 for a derivation).

$$\begin{aligned}\mathcal{L}(\pi_{1:J}) = &\sum_{j=1}^{J}\pi_j \log\left(\sum_{\ell=1}^{J}\pi_\ell \mathcal{N}(\hat{\mathbf{x}}_j; \hat{\mathbf{x}}_\ell, \mathbf{\Sigma}_j + \mathbf{\Sigma}_\ell)\right)\\ &- \sum_{j=1}^{J}\pi_j \log\left(\sum_{n=1}^{N}\pi_{0n}\mathcal{N}(\hat{\mathbf{x}}_j; \hat{\mathbf{x}}_{0n}, \mathbf{\Sigma}_j + \mathbf{\Sigma}_{0n})\right)\\ &- \sum_{j=1}^{J}\pi_j \mathrm{E}_{q_j}\left[\log p(\mathbf{y}|\mathbf{x})\right] + \lambda\left(1 - \sum_{j=1}^{J}\pi_j\right)\end{aligned} \tag{3.24a}$$

$$\begin{aligned}\approx &\sum_{j=1}^{J}\pi_j \log\left(\sum_{\ell\in\mathcal{N}_j}\pi_\ell \mathcal{N}(\hat{\mathbf{x}}_j; \hat{\mathbf{x}}_\ell, \mathbf{\Sigma}_j + \mathbf{\Sigma}_\ell)\right)\\ &- \sum_{j=1}^{J}\pi_j \log\left(\sum_{n=1}^{N}\pi_{0n}\mathcal{N}(\hat{\mathbf{x}}_j; \hat{\mathbf{x}}_{0n}, \mathbf{\Sigma}_j + \mathbf{\Sigma}_{0n})\right)\\ &- \sum_{j=1}^{J}\pi_j \mathrm{E}_{q_j}\left[\log p(\mathbf{y}|\mathbf{x})\right] + \lambda\left(1 - \sum_{j=1}^{J}\pi_j\right)\end{aligned} \tag{3.24b}$$

$$\begin{aligned}= &\sum_{j=1}^{J}\pi_j \log\left(\mathcal{N}(\hat{\mathbf{x}}_j; \hat{\mathbf{x}}_j, 2\mathbf{\Sigma}_j)\sum_{\ell\in\mathcal{N}_j}\pi_\ell\right)\\ &- \sum_{j=1}^{J}\pi_j \log\left(\sum_{n=1}^{N}\pi_{0n}\mathcal{N}(\hat{\mathbf{x}}_j; \hat{\mathbf{x}}_{0n}, \mathbf{\Sigma}_j + \mathbf{\Sigma}_{0n})\right)\\ &- \sum_{j=1}^{J}\pi_j \mathrm{E}_{q_j}\left[\log p(\mathbf{y}|\mathbf{x})\right] + \lambda\left(1 - \sum_{j=1}^{J}\pi_j\right)\end{aligned} \tag{3.24c}$$

$$\stackrel{\pm}{=} -\frac{1}{2}\sum_{j=1}^{J}\pi_j \log|\mathbf{\Sigma}_j| + \sum_{j=1}^{J}\pi_j \log\left(\sum_{\ell\in\mathcal{N}_j}\pi_\ell\right)$$

$$
\begin{aligned}
&-\sum_{j=1}^{J}\pi_j \log\left(\sum_{n=1}^{N}\pi_{0n}\mathcal{N}(\hat{\mathbf{x}}_j;\hat{\mathbf{x}}_{0n},\mathbf{\Sigma}_j+\mathbf{\Sigma}_{0n})\right)\\
&-\sum_{j=1}^{J}\pi_j \,\mathrm{E}_{q_j}\left[\log p(\mathbf{y}|\mathbf{x})\right]+\lambda\left(1-\sum_{j=1}^{J}\pi_j\right).
\end{aligned} \tag{3.24d}
$$

The derivative of the cost in (3.24d) with respect to the weight $\pi_i$ is given as

$$
\begin{aligned}
\nabla_{\pi_i}\mathcal{L}\left(\pi_{1:J}\right) &= 1-\lambda+\log\left(\sum_{j\in\mathcal{N}_i}\pi_j\right)-\mathrm{E}_{q_i}\left[\log p(\mathbf{y}|\mathbf{x})\right]\\
&-\frac{1}{2}\log|\mathbf{\Sigma}_i|-\log\left(\sum_{n=1}^{N}\pi_{0n}\mathcal{N}(\hat{\mathbf{x}}_i;\hat{\mathbf{x}}_{0n},\mathbf{\Sigma}_i+\mathbf{\Sigma}_{0n})\right).
\end{aligned} \tag{3.25}
$$

Solving the equation $\nabla_{\pi_i}\mathcal{L}\left(\pi_{1:J}\right) = 0$ for $\sum_{j\in\mathcal{N}_i}\pi_j$; enforcing the constraint $\sum_{i=1}^{J}\pi_i = 1$ to find and substitute $\lambda$; and then distributing the weights identically for $j \in \mathcal{N}_i$ would result in the following expression.

$$
\pi_i \propto \frac{1}{\#\{\mathcal{N}_i\}}|\mathbf{\Sigma}_i|^{\frac{1}{2}}\exp\left(\mathrm{E}_{q_i}\left[\log p(\mathbf{y}|\mathbf{x})\right]\right)\sum_{n=1}^{N}\pi_{0n}\mathcal{N}(\hat{\mathbf{x}}_i;\hat{\mathbf{x}}_{0n},\mathbf{\Sigma}_i+\mathbf{\Sigma}_{0n}), \tag{3.26}
$$

for $i = 1,\ldots,J$, where the notation $\#\{\mathcal{N}_i\}$ denotes the cardinality of $\mathcal{N}_i$, i.e., the number of components identical to the $i$th component.

### 3.4.3 Approximation of the Expectations

In order to evaluate the right hand sides of the implicit expressions (3.22), (3.19), and (3.26) we need to calculate the expectation $\mathrm{E}_{q_i}[\log p(\mathbf{y}|\mathbf{x})]$, $\mathrm{E}_{q_i}[\nabla_{\mathbf{x}}\log p(\mathbf{y}|\mathbf{x})]$ and $\mathrm{E}_{q_i}[\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathrm{T}}\log p(\mathbf{y}|\mathbf{x})]$ where

$$
\log p(\mathbf{y}|\mathbf{x}) \stackrel{+}{=} -\frac{1}{2}\log|\mathbf{R}|-\frac{1}{2}(\mathbf{y}-\mathbf{h}(\mathbf{x}))^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-\mathbf{h}(\mathbf{x})). \tag{3.27}
$$

There are different ways to evaluate such expectations in the literature, such as standard linearization, statistical linearization, sigma-point and Monte Carlo approaches. In this work, we propose performing statistical linearization because we have observed in our early trials that it has superior performance compared to standard linearization and it is computationally cheaper than Monte Carlo methods. In order to calculate the expectations $\mathrm{E}_{q_i}[\cdot]$, we approximate the nonlinear function $\mathbf{h}(\cdot)$ in $\log p(\mathbf{y}|\mathbf{x})$ in the support of the $i$th posterior mixture component $q_i(\cdot)$ as follows [13].

$$
\mathbf{h}(\mathbf{x}) \approx \mathbf{H}_i\mathbf{x}+\mathbf{g}_i+\mathbf{e}_i, \tag{3.28}
$$

where $\mathbf{H}_i$ and $\mathbf{g}_i$ are the gain matrix and the bias vector and $\mathbf{e}_i \in \mathbb{R}^{n_y}$ is a Gaussian random vector, which is uncorrelated with $\mathbf{x}$ and the measurement noise $\mathbf{v}$, with zero-mean and covariance $\mathbf{\Omega}_i$. The quantities $\mathbf{H}_i$ and $\mathbf{g}_i$ are found by solving the following optimization problem [12, Section 5.3].

$$\{\mathbf{H}_i^*, \mathbf{g}_i^*\} \triangleq \arg\min_{\mathbf{H}_i, \mathbf{g}_i} \mathrm{E}_{q_i}\left[\|\mathbf{h}(\mathbf{x}) - \mathbf{H}_i\mathbf{x} - \mathbf{g}_i\|^2\right], \tag{3.29}$$

which has the solution

$$\mathbf{H}_i = \mathrm{E}_{q_i}\left[\big(\mathbf{h}(\mathbf{x}) - \mathrm{E}_{q_i}[\mathbf{h}(\mathbf{x})]\big)(\mathbf{x} - \hat{\mathbf{x}}_i)^{\mathrm{T}}\right]\mathbf{\Sigma}_i^{-1}, \tag{3.30a}$$

$$\mathbf{g}_i = \mathrm{E}_{q_i}\left[\mathbf{h}(\mathbf{x})\right] - \mathbf{H}_i\hat{\mathbf{x}}_i. \tag{3.30b}$$

The covariance $\mathbf{\Omega}_i$ is selected as [13]

$$\mathbf{\Omega}_i = \mathrm{E}_{q_i}\left[\big(\mathbf{h}(\mathbf{x}) - \mathrm{E}_{q_i}[\mathbf{h}(\mathbf{x})]\big)\big(\mathbf{h}(\mathbf{x}) - \mathrm{E}_{q_i}[\mathbf{h}(\mathbf{x})]\big)^{\mathrm{T}}\right] - \mathbf{H}_i\mathbf{\Sigma}_i\mathbf{H}_i^{\mathrm{T}}. \tag{3.31}$$

The expectations in (3.30) and (3.31) are taken using sigma-point approaches.

With the approximation (3.28), the measurement log-likelihood $\log p(\mathbf{y}|\mathbf{x})$ in (3.27) turns into the following expression.

$$\log p(\mathbf{y}|\mathbf{x}) \stackrel{+}{=} -\frac{1}{2}\log|\mathbf{R} + \mathbf{\Omega}_i| - \frac{1}{2}(\mathbf{y} - \mathbf{H}_i\mathbf{x} - \mathbf{g}_i)^{\mathrm{T}}(\mathbf{R} + \mathbf{\Omega}_i)^{-1}(\mathbf{y} - \mathbf{H}_i\mathbf{x} - \mathbf{g}_i). \tag{3.32}$$

We can then calculate

$$\begin{aligned}\mathrm{E}_{q_i}[\log p(\mathbf{y}|\mathbf{x})] \stackrel{+}{=}& -\frac{1}{2}\log|\mathbf{R} + \mathbf{\Omega}_i| \\ &- \frac{1}{2}(\mathbf{y} - \mathbf{H}_i\hat{\mathbf{x}}_i - \mathbf{g}_i)^{\mathrm{T}}(\mathbf{R} + \mathbf{\Omega}_i)^{-1}(\mathbf{y} - \mathbf{H}_i\hat{\mathbf{x}}_i - \mathbf{g}_i) \\ &- \frac{1}{2}\operatorname{tr}\big(\mathbf{H}_i^{\mathrm{T}}(\mathbf{R} + \mathbf{\Omega}_i)^{-1}\mathbf{H}_i\mathbf{\Sigma}_i\big),\end{aligned} \tag{3.33a}$$

$$\mathrm{E}_{q_i}\left[\nabla_{\mathbf{x}}\log p(\mathbf{y}|\mathbf{x})\right] = \mathbf{H}_i^{\mathrm{T}}(\mathbf{R} + \mathbf{\Omega}_i)^{-1}(\mathbf{y} - \mathbf{H}_i\hat{\mathbf{x}}_i - \mathbf{g}_i), \tag{3.33b}$$

$$\mathrm{E}_{q_i}\left[\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathrm{T}}\log p(\mathbf{y}|\mathbf{x})\right] = -\mathbf{H}_i^{\mathrm{T}}(\mathbf{R} + \mathbf{\Omega}_i)^{-1}\mathbf{H}_i. \tag{3.33c}$$

When we substitute the approximate expectations in (3.33) into (3.19), (3.22) and (3.26), we get the following final implicit expressions for the covariance $\mathbf{\Sigma}_i$, the mean $\hat{\mathbf{x}}_i$ and the weight $\pi_i$, respectively.

$$\mathbf{\Sigma}_i = \left(\sum_{n=1}^{N}\beta_{i,n}\mathbf{\Sigma}_{0n}^{-1} + \mathbf{H}_i^{\mathrm{T}}(\mathbf{R} + \mathbf{\Omega}_i)^{-1}\mathbf{H}_i\right)^{-1}, \tag{3.34a}$$

$$\hat{\mathbf{x}}_i = \hat{\mathbf{x}}_i + \alpha\boldsymbol{\Sigma}_i\mathbf{H}_i^{\mathrm{T}}\left(\mathbf{R}+\boldsymbol{\Omega}_i\right)^{-1}\left(\mathbf{y}-\mathbf{H}_i\hat{\mathbf{x}}_i-\mathbf{g}_i\right)+\alpha\boldsymbol{\Sigma}_i\sum_{j=1}^{J}\gamma_{i,j}\boldsymbol{\Sigma}_j^{-1}\left(\hat{\mathbf{x}}_i-\hat{\mathbf{x}}_j\right)$$

$$-\alpha\boldsymbol{\Sigma}_i\sum_{n=1}^{N}\beta_{i,n}\boldsymbol{\Sigma}_{0n}^{-1}(\hat{\mathbf{x}}_i-\hat{\mathbf{x}}_{0n}), \tag{3.34b}$$

$$\pi_i \propto \frac{1}{\#\{\mathcal{N}_i\}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}\exp\Big(-\frac{1}{2}\operatorname{tr}\left(\mathbf{H}_i^{\mathrm{T}}(\mathbf{R}+\boldsymbol{\Omega}_i)^{-1}\mathbf{H}_i\boldsymbol{\Sigma}_i\right)\Big)\mathcal{N}(\mathbf{y};\mathbf{H}_i\hat{\mathbf{x}}_i+\mathbf{g}_i,\mathbf{R}+\boldsymbol{\Omega}_i)$$

$$\times\sum_{n=1}^{N}\pi_{0n}\mathcal{N}(\hat{\mathbf{x}}_i;\hat{\mathbf{x}}_{0n},\boldsymbol{\Sigma}_i+\boldsymbol{\Sigma}_{0n}), \tag{3.34c}$$

for $i = 1,\ldots,J$.

Note that the expectation $\mathrm{E}_{q_i}\left[\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathrm{T}}\log p(\mathbf{y}|\mathbf{x})\right]$ has connections to the concept of observed information matrix (OIM) described in [48] given as

$$\mathbf{J}^O = -\left[\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathrm{T}}\log p(\mathbf{y},\mathbf{x})\right]\big|_{\mathbf{x}=\hat{\mathbf{x}}_{map}}, \tag{3.35}$$

where $\hat{\mathbf{x}}_{map}$ is the maximum a posteriori (MAP) estimate of $\mathbf{x}$. An alternative calculation/approximation for $\mathrm{E}_{q_i}\left[\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathrm{T}}\log p(\mathbf{y}|\mathbf{x})\right]$ and hence for $\boldsymbol{\Sigma}_i$ can be developed by writing $\log p(\mathbf{y},\mathbf{x})$ as follows:

$$\log p(\mathbf{y},\mathbf{x}) = \log p(\mathbf{y}|\mathbf{x}) + \log p(\mathbf{x}). \tag{3.36}$$

Taking the Hessian and the expected value of both sides with respect to $\mathbf{x}$ we get

$$\mathrm{E}_{q_i}\left[\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathrm{T}}\log p(\mathbf{y},\mathbf{x})\right] = \mathrm{E}_{q_i}\left[\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathrm{T}}\log p(\mathbf{y}|\mathbf{x})\right] + \mathrm{E}_{q_i}\left[\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathrm{T}}\log p(\mathbf{x})\right], \tag{3.37}$$

which gives

$$\mathrm{E}_{q_i}\left[\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathrm{T}}\log p(\mathbf{y}|\mathbf{x})\right] = \mathrm{E}_{q_i}\left[\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathrm{T}}\log p(\mathbf{y},\mathbf{x})\right] - \mathrm{E}_{q_i}\left[\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathrm{T}}\log p(\mathbf{x})\right]. \tag{3.38}$$

We can now approximate $\mathrm{E}_{q_i}\left[\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathrm{T}}\log p(\mathbf{y},\mathbf{x})\right]$ as

$$\mathrm{E}_{q_i}\left[\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathrm{T}}\log p(\mathbf{y},\mathbf{x})\right] \approx \nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathrm{T}}\log p(\mathbf{y},\mathbf{x})\big|_{\mathbf{x}=\hat{\mathbf{x}}_{map}} \tag{3.39a}$$

$$= -\mathbf{J}^O, \tag{3.39b}$$

which is the so-called Laplace approximation [49, Chapter 27]. Substituting approximation (3.39) into (3.38) we get

$$\mathrm{E}_{q_i}\left[\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathrm{T}}\log p(\mathbf{y}|\mathbf{x})\right] \approx -\mathbf{J}^O + \sum_{n=1}^{N}\beta_{i,n}\boldsymbol{\Sigma}_{0n}^{-1}, \tag{3.40}$$

where we used (3.73). When we substitute (3.40) into (3.19), we get

$$\mathbf{\Sigma}_i \approx \left(\mathbf{J}^O\right)^{-1}. \tag{3.41}$$

As a result, an alternative update rule for the covariance $\mathbf{\Sigma}_i$ is to equate it to inverse of the OIM. Note that OIM calculation requires another optimization to find $\hat{\mathbf{x}}_{map}$ which would increase the computational load significantly. As a result, we will not follow this methodology and continue to use (3.34a) in the following.

### 3.4.4 Iterative Measurement Update

We can use the implicit expressions in (3.34) to form an iterative procedure to minimize $L(q)$ in (3.8). In order to do this, we can first set the unknown posterior $q(\mathbf{x})$ to an initial mixture $q^{(0)}(\mathbf{x}) = \sum_{j=1}^{J} \pi_j^{(0)} \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_j^{(0)}, \mathbf{\Sigma}_j^{(0)})$. Now, given the last posterior estimate $q^{(m-1)}(\mathbf{x}) = \sum_{j=1}^{J} \pi_j^{(m-1)} \mathcal{N}\big(\mathbf{x}; \hat{\mathbf{x}}_j^{(m-1)}, \mathbf{\Sigma}_j^{(m-1)}\big)$, we can use the implicit expressions in (3.34) to find the next posterior estimate $q^{(m)}(\mathbf{x}) = \sum_{j=1}^{J} \pi_j^{(m)} \mathcal{N}\big(\mathbf{x}; \hat{\mathbf{x}}_j^{(m)}, \mathbf{\Sigma}_j^{(m)}\big)$, for $m \geq 1$ until convergence. For this purpose, we can substitute the quantities $\pi_{1:J}^{(m-1)}, \hat{\mathbf{x}}_{1:J}^{(m-1)}, \mathbf{\Sigma}_{1:J}^{(m-1)}$ (and the corresponding derived quantities $\mathbf{H}_{1:J}, \mathbf{g}_{1:J}, \mathbf{\Omega}_{1:J}$) on the right hand sides of the implicit expressions in (3.34) to calculate the quantities $\pi_{1:J}^{(m)}, \hat{\mathbf{x}}_{1:J}^{(m)}, \mathbf{\Sigma}_{1:J}^{(m)}$.

The pseudo-code of the iterative measurement update we propose is given in Algorithm 1. The main difference of the proposed algorithm from the description above is that when a quantity is updated to its current version, its updated version is used for updating the subsequent quantities. For example, when we update $\hat{\mathbf{x}}_1^{(m-1)}$ to its current iterate $\hat{\mathbf{x}}_1^{(m)}$, we use the updated iterate $\hat{\mathbf{x}}_1^{(m)}$ (instead of the previous iterate $\hat{\mathbf{x}}_1^{(m-1)}$) in the update of $\hat{\mathbf{x}}_{2:J}$ from $\hat{\mathbf{x}}_{2:J}^{(m-1)}$ to $\hat{\mathbf{x}}_{2:J}^{(m)}$. We have seen that this type of update structure significantly reduces the time required for the convergence compared to the description in the first paragraph above where the previous iterates of the quantities $\pi_{1:J}^{(m-1)}, \hat{\mathbf{x}}_{1:J}^{(m-1)}, \mathbf{\Sigma}_{1:J}^{(m-1)}$ are used in the update of all quantities $\pi_{1:J}, \hat{\mathbf{x}}_{1:J}, \mathbf{\Sigma}_{1:J}$ into their current iterates $\pi_{1:J}^{(m)}, \hat{\mathbf{x}}_{1:J}^{(m)}, \mathbf{\Sigma}_{1:J}^{(m)}$. The structure and the faster converge of the proposed update structure compared to the description in the first paragraph above is, in a sense, similar to the structure and faster convergence of Gauss-Seidel method compared to the Jacobi method for solving linear systems of equations.

**Algorithm 1** Iterative Measurement Update

**Inputs:** $\mathbf{y}, \mathbf{R}, \{\pi_{0n}, \hat{\mathbf{x}}_{0n}, \mathbf{\Sigma}_{0n}\}_{n=1}^{N}, \alpha, \tau_C, \eta_{\mathcal{N}}, M$

Initialize $\hat{\mathbf{x}}_{1:J}^{(0)}$, $\mathbf{\Sigma}_{1:J}^{(0)}$ using $\mathbf{y}$, $\mathbf{R}$, $\{\pi_{0n}, \hat{\mathbf{x}}_{0n}, \mathbf{\Sigma}_{0n}\}_{n=1}^{N}$.

Set $\pi_i^{(0)} = 1/J$, $i = 1, \ldots, J$.

Set $m = 0$.

**repeat**

    $m \leftarrow m + 1$

    $\bar{\mathbf{x}}_i = \hat{\mathbf{x}}_i^{(m-1)}$, $i = 1, \ldots, J$.

    **for** $i = 1$ to $J$ **do**

        Generate sigma points $\mathbf{s}_{1:L}$ and their weights $w_{1:L}$ from $\hat{\mathbf{x}}_i^{(m-1)}$, $\mathbf{\Sigma}_i^{(m-1)}$.

        Calculate $\mathbf{H}_i$, $\mathbf{g}_i$, $\mathbf{\Omega}_i$ using $w_{1:L}$, $\mathbf{s}_{1:L}$, $\hat{\mathbf{x}}_i^{(m-1)}$, $\mathbf{\Sigma}_i^{(m-1)}$ via (3.30) and (3.31).

        Calculate $\gamma_{i,1:J}$ using $\pi_{1:J}^{(m-1)}$, $\hat{\mathbf{x}}_{1:i-1}^{(m)}$, $\hat{\mathbf{x}}_{i:J}^{(m-1)}$, $\mathbf{\Sigma}_{1:i-1}^{(m)}$, $\mathbf{\Sigma}_{i:J}^{(m-1)}$ via (3.10a).

        Calculate $\beta_{i,1:N}$ using $\hat{\mathbf{x}}_i^{(m-1)}$ via (3.10b).

        Calculate the covariance $\mathbf{\Sigma}_i^{(m)}$ using $\beta_{i,1:N}$, $\mathbf{H}_i$, $\mathbf{\Omega}_i$ via (3.34a).

        Calculate the mean $\hat{\mathbf{x}}_i^{(m)}$ using $\mathbf{H}_i$, $\mathbf{g}_i$, $\mathbf{\Omega}_i$, $\beta_{i,1:N}$, $\gamma_{i,1:J}$, $\hat{\mathbf{x}}_{1:i-1}^{(m)}$, $\hat{\mathbf{x}}_{i:J}^{(m-1)}$, $\mathbf{\Sigma}_{1:i}^{(m)}$, $\mathbf{\Sigma}_{i+1:J}^{(m-1)}$ via (3.34b).

    **end for**

    **for** $i = 1$ to $J$ **do**

        $\#\{\mathcal{N}_i\} = 0$ ▷ Number of Identical Components

        **for** $j = 1$ to $J$ **do**

            $\eta = (\hat{\mathbf{x}}_i^{(m)} - \hat{\mathbf{x}}_j^{(m)})^{\mathrm{T}}(\mathbf{\Sigma}_i^{(m)} + \mathbf{\Sigma}_j^{(m)})^{-1} \times (\hat{\mathbf{x}}_i^{(m)} - \hat{\mathbf{x}}_j^{(m)})$

            **if** $\eta < \eta_{\mathcal{N}}$ **then**

                $\#\{\mathcal{N}_i\} = \#\{\mathcal{N}_i\} + 1$

            **end if**

        **end for**

        Calculate the weight $\pi_i^{(m)}$ using $\#\{\mathcal{N}_i\}$, $\mathbf{H}_i$, $\mathbf{g}_i$,

$\boldsymbol{\Omega}_i$, $\hat{\mathbf{x}}_i^{(m)}$, $\boldsymbol{\Sigma}_i^{(m)}$ via (3.34c).

**end for**

Normalize $\pi_{1:J}^{(m)}$.

$\tau = \max_{i \in \{1,\ldots,J\}} \|\hat{\mathbf{x}}_i^{(m)} - \bar{\mathbf{x}}_i\|_2$

**until** $\tau < \tau_C$ or $m \geq M$ ▷ Convergence

**Outputs:** $\pi_{1:J}^{(m)}, \hat{\mathbf{x}}_{1:J}^{(m)}, \boldsymbol{\Sigma}_{1:J}^{(m)}$

### 3.4.5 Special Case $N = J = 1$

When we set the number of mixture components $N$ and $J$ as $N = J = 1$, the prior and posterior mixtures turn into Gaussian distributions, i.e., $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_0, \boldsymbol{\Sigma}_0)$ and $q(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \boldsymbol{\Sigma})$. In this case the implicit expressions in (3.34a) and (3.34b) reduce to

$$\boldsymbol{\Sigma} = \left(\boldsymbol{\Sigma}_0^{-1} + \mathbf{H}^{\mathrm{T}} \left(\mathbf{R} + \boldsymbol{\Omega}\right)^{-1} \mathbf{H}\right)^{-1}, \tag{3.42a}$$

$$\hat{\mathbf{x}} = \hat{\mathbf{x}} + \alpha \boldsymbol{\Sigma} \mathbf{H}^{\mathrm{T}} \left(\mathbf{R} + \boldsymbol{\Omega}\right)^{-1} \left(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}} - \mathbf{g}\right) - \alpha \boldsymbol{\Sigma} \boldsymbol{\Sigma}_0^{-1} (\hat{\mathbf{x}} - \hat{\mathbf{x}}_0), \tag{3.42b}$$

where $\mathbf{H}$, $\mathbf{g}$ and $\boldsymbol{\Omega}$ are calculated using statistical linearization based on (3.30) and (3.31) using the posterior mean $\hat{\mathbf{x}}$ and covariance $\boldsymbol{\Sigma}$. We can write the iterative update corresponding to the implicit expressions in (3.42) by following the update structure of Algorithm 1 as follows.

$$\boldsymbol{\Sigma}^{(m)} = \left(\boldsymbol{\Sigma}_0^{-1} + \mathbf{H}^{\mathrm{T}} \left(\mathbf{R} + \boldsymbol{\Omega}\right)^{-1} \mathbf{H}\right)^{-1}, \tag{3.43a}$$

$$\begin{aligned} \hat{\mathbf{x}}^{(m)} = {} & \hat{\mathbf{x}}^{(m-1)} + \alpha \boldsymbol{\Sigma}^{m} \mathbf{H}^{\mathrm{T}} \left(\mathbf{R} + \boldsymbol{\Omega}\right)^{-1} \left(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}^{(m-1)} - \mathbf{g}\right) \\ & - \alpha \boldsymbol{\Sigma}^{(m)} \boldsymbol{\Sigma}_0^{-1} (\hat{\mathbf{x}}^{(m-1)} - \hat{\mathbf{x}}_0), \end{aligned} \tag{3.43b}$$

where $\mathbf{H}$, $\mathbf{g}$ and $\boldsymbol{\Omega}$ are calculated using statistical linearization based on (3.30) and (3.31) using the previous posterior mean iterate $\hat{\mathbf{x}}^{(m-1)}$ and covariance iterate $\boldsymbol{\Sigma}^{(m-1)}$. The expressions in (3.43) are the update equations for the posterior linearization filter (PLF) [13] when $\alpha = 1$ and the damped PLF [14] when $\alpha < 1$. Hence we see that our iterative update reduces to PLF when $N = J = 1$ and, in a sense, PLF minimizes forward Kullback-Leibler divergence as well.

### 3.5 Illustration on a Static Bayesian Estimation Problem

In this section we illustrate the characteristics of the measurement update we propose on a common static Bayesian estimation problem. For this purpose we consider a 2D target localization problem based on the noisy range-bearing data collected from a single sensor. The unknown 2D target location to be estimated is denoted as $\mathbf{x} \triangleq [p_\mathsf{x} \;\; p_\mathsf{y}]^\mathrm{T}$ and it has the following Gaussian prior.

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_0, \mathbf{P}_0) \tag{3.44}$$

where $\hat{\mathbf{x}}_0 = [5 \;\; 0]^\mathrm{T}$ and $\mathbf{P}_0 = 10^2 \mathbf{I}_2$. The sensor is located at the origin and it collects range and bearing measurement $\mathbf{y}$ described by the measurement function $\mathbf{h}(\mathbf{x})$ and covariance $\mathbf{R}$ given as

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} (p_\mathsf{x}^2 + p_\mathsf{y}^2)^{1/2} \\ \arctan \frac{p_\mathsf{y}}{p_\mathsf{x}} \end{bmatrix}, \qquad \mathbf{R} = \begin{bmatrix} 4 & 0 \\ 0 & (30\pi/180)^2 \end{bmatrix}. \tag{3.45}$$

We assume that we collect the single measurement $\mathbf{y} = [20 \;\; \pi/2]^\mathrm{T}$ from the sensor and would like to estimate the posterior distribution $p(\mathbf{x}|\mathbf{y})$ of the target location $\mathbf{x}$.

We implemented the following algorithms to obtain the posterior.

- **PF:** Bootstrap particle filter [24] with the number of particles $10^5$ whose result is going to serve as the true posterior.
- **EKF:** Extended Kalman filter [50].
- **UKF:** Unscented Kalman filter [1]. 5 sigma-points are used. The weight of the sigma-point at the mean is taken to be 0.5.
- **IEKF:** Iterated extended Kalman filter [8].
- **PLF:** Posterior linearization filter [13].
- **FKLF-$J$:** The proposed algorithm with the step size $\alpha = 0.1$ and the number of components $J = 1, \ldots, 6$.

The corresponding estimated posteriors are illustrated in Figure 3.1. EKF and UKF, which apply linearization and statistical linearization, respectively, based on the prior, yield quite poor results considering the true posterior. The iterative nature of IEKF seems to have fixed the result of EKF to some extent. It is seen that PLF and FKLF-1 indeed give the same Gaussian posterior which has closer mean and covariance to those of the true posterior than the other methods providing Gaussian posteriors.

When the number of components $J$ in FKLF is increased, the Gaussian components are observed to spread to cover a larger part of the support of the true posterior. In Figures 3.1d, 3.1e and 3.1f, we see that although FKLF is run with 4, 5 and 6 components, respectively, some components turned out to be the same, effectively reducing the number of components of the resulting posterior to 3, 4 and 4, respectively. When we compare the results of Figures 3.1c and 3.1d we see that both posteriors provided by FKLF have 3 distinct components and the three components obtained by FKLF are slightly different from each other since the support of the posterior in Figure 3.1d has marginally shifted to the left. This shows that even when we constrain the result to have $J$ distinct components, there might be multiple local optima of the optimization problem. In Figures 3.1e and 3.1f we see that the same mixtures have been obtained with different sets of identical components. It is seen in the figures that the overall means and covariances of the posteriors obtained by FKLF-$J$, $J = 2, \ldots, 6$, are very close to those of the true posterior.

Table 3.1: Average simulation times of the algorithms in milliseconds.

| FKLF-1 | FKLF-2 | FKLF-3 | FKLF-4 | FKLF-5 | FKLF-6 | PF |
|---|---|---|---|---|---|---|
| 2.6 | 17 | 64.2 | 124.2 | 208.1 | 379.1 | 36.9 |

To investigate the simulation times of the proposed solution and the particle filter, we make 1000 Monte Carlo runs with different measurements. These measurements are obtained by adding Gaussian noise having zero mean and the covariance matrix $\mathbf{R}$ to the vector $\mathbf{y}$. The simulations are carried out on a computer with a CPU speed of 2.1 GHz. Average simulation times for a single run are provided in Table 3.1.

Table 3.1 indicates that increasing the number of components in the suggested solution increases the simulation times. When the number of components in the FKLF solution exceeds two, the simulation time for the FKLF exceeds the simulation time

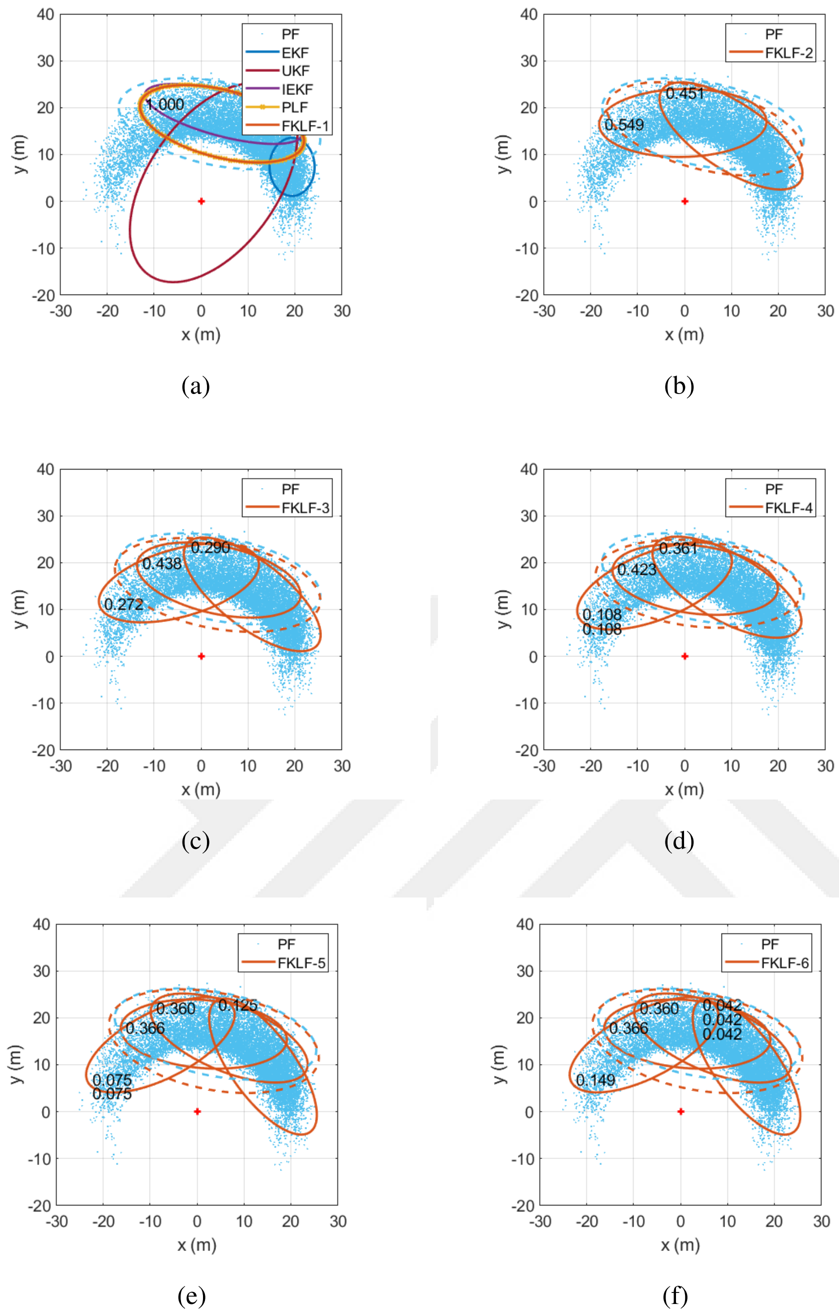



Figure 3.1: The estimated posterior densities for the static Bayesian estimation problem. The light blue dots represent the resampled particles of the PF. Gaussian densities are illustrated with 95% confidence ellipses (solid lines) of their covariances centered around their means. Ellipses in dashed lines represent the mean and covariance of the posteriors obtained by PF and FKLF. The weights of the Gaussian components of the posterior obtained by FKLF are written at the left vertex along the semi-major axis of the confidence ellipse of each component. The sensor position is shown with a plus sign in red.

for the particle filter. We should note that we have two dimensional state in the experiment. When this dimension is high, the PF would require a large number of particles, which would increase its simulation time significantly.

## 3.6 Numerical Results

### 3.6.1 Scenario

We test the performance of the proposed filter on the 2D target tracking problem given in [13], where there is a single target moving with nearly constant velocity in a sensor network. The state $\mathbf{x} \triangleq [\,\mathrm{x} \;\; \dot{\mathrm{x}} \;\; \mathrm{y} \;\; \dot{\mathrm{y}}\,]^{\mathrm{T}}$ of the target is composed of 2D position and velocity of the target and it evolves according to the nearly constant velocity model given below.

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_k, \tag{3.46}$$

for $k = 1, \ldots, 19$, where

$$\mathbf{F} = \mathbf{I}_2 \otimes \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \qquad \Delta t = 0.5\,\mathrm{s}, \tag{3.47}$$

and $\mathbf{w}_k$ is white Gaussian process noise with zero mean and covariance matrix $\mathbf{Q}$ defined as

$$\mathbf{Q} = q\mathbf{I}_2 \otimes \begin{bmatrix} \Delta t^3/3 & \Delta t^2/2 \\ \Delta t^2/2 & \Delta t \end{bmatrix}, \qquad q = 1\,\mathrm{m}^2/\mathrm{s}^3. \tag{3.48}$$

The initial state $\mathbf{x}_0$ of the target is distributed as $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{x}_0; \bar{\mathbf{x}}_0, \boldsymbol{\Sigma}_0)$ where

$$\bar{\mathbf{x}}_0 = [50 \;\; 0 \;\; 50 \;\; 0]^{\mathrm{T}}, \qquad \boldsymbol{\Sigma}_0 \triangleq \mathbf{I}_2 \otimes \begin{bmatrix} 25 & 0 \\ 0 & 4 \end{bmatrix}. \tag{3.49}$$

We show 1000 random sample trajectories generated from this mathematical model in Figure 3.2.

The sensor network, shown in Figure 3.2, is composed of 25 sensors placed on a grid of separation 25 m. Each of the sensors measure the acoustic signal strength emitted by the target which is modeled as

$$y_k^s = h_s(\mathbf{x}_k) + v_k^s, \tag{3.50}$$

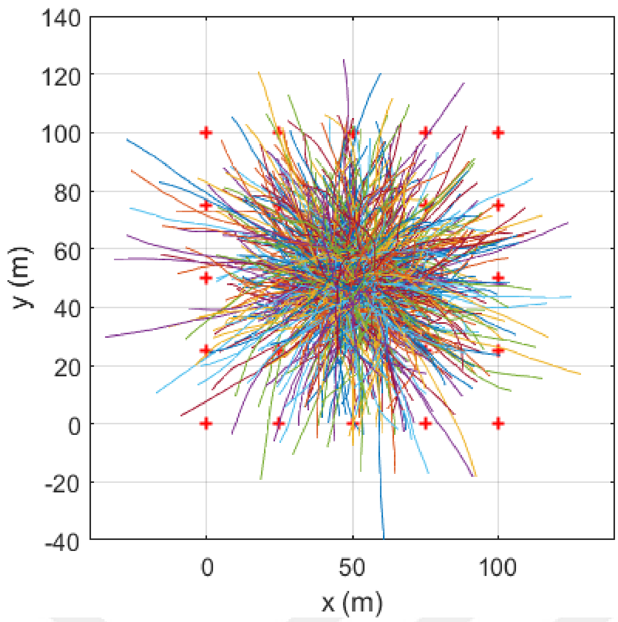



Figure 3.2: Sensor positions (red plus signs) and 1000 random sample target trajectories used in the simulations.

where $s = 1, \dots, 25$, denotes the sensor index; $v_k^s$ is the white Gaussian measurement noise with zero mean and variance $\sigma_v^2 = 0.1\,\text{W}$; and $h_s(\mathbf{x}_k)$ is defined as

$$h_s(\mathbf{x}) \triangleq \begin{cases} \sqrt{\frac{P_0 d_0^2}{d_s^2(\mathbf{x})}}, & d_s^2(\mathbf{x}) > d_0^2 \\ \sqrt{P_0}, & d_s^2(\mathbf{x}) \leq d_0^2 \end{cases}, \tag{3.51}$$

where $P_0 = 1000\,\text{W}$ is the saturation power received at the fixed distance $d_0 = 1\,\text{m}$ and $d_s(\mathbf{x})$ denotes the Euclidean distance between the $s$th sensor and the target. Defining the overall measurement vector $\mathbf{y}_k$, the measurement noise vector $\mathbf{v}_k$ and the measurement function $\mathbf{h}(\mathbf{x})$ as

$$\mathbf{y}_k \triangleq [\, y_k^1 \;\; y_k^2 \;\; \cdots \;\; y_k^{25} \,]^{\mathrm{T}}, \tag{3.52a}$$

$$\mathbf{v}_k \triangleq [\, v_k^1 \;\; v_k^2 \;\; \cdots \;\; v_k^{25} \,]^{\mathrm{T}}, \tag{3.52b}$$

$$\mathbf{h}(\mathbf{x}) \triangleq [\, h_1(\mathbf{x}) \;\; h_2(\mathbf{x}) \;\; \cdots \;\; h_{25}(\mathbf{x}) \,]^{\mathrm{T}}, \tag{3.52c}$$

we can write the measurement model given below.

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k, \tag{3.53}$$

where $\mathbf{v}_k \sim \mathcal{N}(\mathbf{v}_k; \mathbf{0}, \sigma_v^2 \mathbf{I}_{25})$ assuming that the measurement noises $v_k^s$, $s = 1, \dots, 25$, are independent.

### 3.6.2 Implemented Methods

A total of 10000 Monte Carlo runs, with a different realization of the state and measurements in each run, have been made by running following algorithms.

- **PF:** Bootstrap particle filter [24] with $10^5$ particles.
- **EKF:** Extended Kalman filter [50].
- **UKF:** Unscented Kalman filter [1].
- **IEKF:** Iterated extended Kalman filter [51] with step-size $\alpha = 0.1$. The standard IEKF (i.e., with step-size $\alpha = 1$ [52]) diverged in many of the Monte-Carlo runs and hence, it is not included in the comparisons. The iterations are assumed to have converged if the $\ell_2$ norm $\|\hat{\mathbf{x}}_k^{(m)} - \hat{\mathbf{x}}_k^{(m-1)}\|_2$ of the difference between two consecutive mean estimates is below the threshold $\tau_C = 10^{-3}$.
- **PLF:** Damped posterior linearization filter [13, 14] with step-size $\alpha = 0.1$. The standard PLF with step-size $\alpha = 1$ diverged in many of the Monte-Carlo runs and hence, it is not included in the comparisons. At each time step, PLF iterations are initialized with the mean and covariance generated by updating the predicted mean and covariance with UKF using the current measurement $\mathbf{y}_k$. The iterations are assumed to have converged if the $\ell_2$ norm $\|\hat{\mathbf{x}}_k^{(m)} - \hat{\mathbf{x}}_k^{(m-1)}\|_2$ of the difference between two consecutive mean estimates is below the threshold $\tau_C = 10^{-3}$.
- **GSF-$J$:** Gaussian sum filter [18] with the number of components $J = 2, 4, 8$. The component updates were performed using UKF. GSF performance was sensitive to the initial prior mixture. Since we have the Gaussian prior $\mathcal{N}(\mathbf{x}_0; \bar{\mathbf{x}}_0, \mathbf{\Sigma}_0)$, an idea is to start GSF with a mixture with identical components, i.e., with the mixture

$$p(\mathbf{x}_0) = \sum_{j=1}^{J} \frac{1}{J} \mathcal{N}(\mathbf{x}_0; \bar{\mathbf{x}}_0^j, \mathbf{\Sigma}_0^j). \tag{3.54}$$

  where $\bar{\mathbf{x}}_0^j = \bar{\mathbf{x}}_0$, $\mathbf{\Sigma}_0^j = \mathbf{\Sigma}_0$ for $j = 1, \ldots, J$. However, in this case, all GSF components remain the same at all time instants and GSF-$J$ turns into UKF. Another idea is to approximate the initial Gaussian prior $\mathcal{N}(\mathbf{x}_0; \bar{\mathbf{x}}_0, \mathbf{\Sigma}_0)$ with a moment-matched Gaussian mixture with smaller covariances, i.e., with a mixture in the form

$$p(\mathbf{x}_0) = \sum_{j=1}^{J} w_j \mathcal{N}(\mathbf{x}_0; \bar{\mathbf{x}}_0^j, \boldsymbol{\Sigma}_0^j). \tag{3.55}$$

where the parameters $w_{1:J}, \bar{\mathbf{x}}_0^{1:J}, \boldsymbol{\Sigma}_0^{1:J}$ are selected deterministically such that the mean of the mixture is $\bar{\mathbf{x}}_0$ and the covariance is $\boldsymbol{\Sigma}_0$. The last idea we tried was to initialize GSF with a random mixture in the form (3.55) where the parameters $w_{1:J}, \bar{\mathbf{x}}_0^{1:J}, \boldsymbol{\Sigma}_0^{1:J}$ are selected randomly such that the expected mean of the mixture is $\bar{\mathbf{x}}_0$ and the expected covariance is $\boldsymbol{\Sigma}_0$. In our experiments we have seen that initialization with the random mixture approach worked significantly better than initialization with the deterministic mixture approach. Consequently, in each Monte Carlo run we initialized GSF-$J$ with a new realization of the Gaussian mixture $p(\mathbf{x}_0)$ in the form (3.55) where the parameters $w_{1:J}, \bar{\mathbf{x}}_0^{1:J}, \boldsymbol{\Sigma}_0^{1:J}$ are selected as follows.

$$w_j = \frac{1}{J}, \qquad \bar{\mathbf{x}}_0^j \sim \mathcal{N}\left(\bar{\mathbf{x}}_0, \frac{J\boldsymbol{\Sigma}_0}{2J-1}\right), \qquad \boldsymbol{\Sigma}_0^j = \frac{J\boldsymbol{\Sigma}_0}{2J-1}, \tag{3.56}$$

for $j = 1, \ldots, J$. Note that with this selection, the expected mean and covariance of the random mixture $p(\mathbf{x}_0)$ becomes $\bar{\mathbf{x}}_0$ and $\boldsymbol{\Sigma}_0$, respectively, whose proof is given in Appendix 3.8.3.

- **FKLF-$J$:** The proposed algorithm with step-size $\alpha = 0.1$ and with the number of components $J = 2, 4, 8$. After each time update, the components in the iterative measurement update are initialized with the means $\hat{\mathbf{x}}_{1:J}^{(0)}$ and covariances $\boldsymbol{\Sigma}_{1:J}^{(0)}$ generated by updating the components of the predicted mixture $p(\mathbf{x}_k|\mathbf{y}_{0:k-1})$ with UKF using the current measurement $\mathbf{y}_k$. The mixture components at each iteration are considered to be identical if the Mahalanobis distance given in line 19 of Algorithm 1 is smaller than the threshold $\eta_{\mathcal{N}} = n_x/1000$ where $n_x = 4$. The convergence threshold and the maximum number of iterations are selected as $\tau_C = 10^{-3}$ and $M = 2000$.

### 3.6.3 Results

We compare the filters in terms of their root mean square (RMS) position errors, RMS velocity errors, and normalized estimation error squared (NEES) values.[4] In addition, we examine the average number of iterations that filters perform at each time step.

RMS position and velocity errors of the filters are shown in Figures 3.3 and 3.4, respectively. In both figures, PF obtains the best performance. The worst performances are obtained by EKF and UKF, the former being worse than the latter. PLF and IEKF are the best and the second best methods among the filters providing a Gaussian posterior. The performance of GSF is poor for $J = 2$ and $4$ while its performance can barely reach the performance of PLF when $J = 8$. FKLF provides the closest results to those of PF. It is seen that increasing the number of components in FKLF reduces RMSE and FKLF-8 asymptotically reaches the performance of PF.

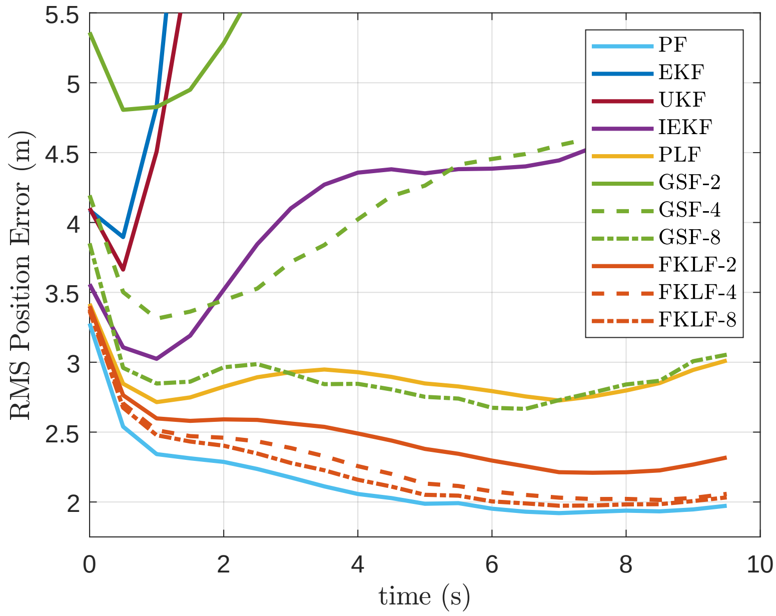



Figure 3.3: Position RMSE performance of the filters.

NEES values of the filters are shown in Figure 3.5. We see that all filters except PF are

[4] NEES for a filter is calculated by taking the average of the normalized estimation error $(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})^{\mathrm{T}} \mathbf{\Sigma}_{k|k}^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})$ over all Monte Carlo runs, where $\mathbf{x}_k$ is the true state and $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{\Sigma}_{k|k}$ are the mean and the covariance of the posterior calculated by the filter, respectively, in the Monte Carlo run. The filter is deemed to have better consistency characteristics if its NEES values are close to $n_x = 4$, i.e., the dimension of the state vector.

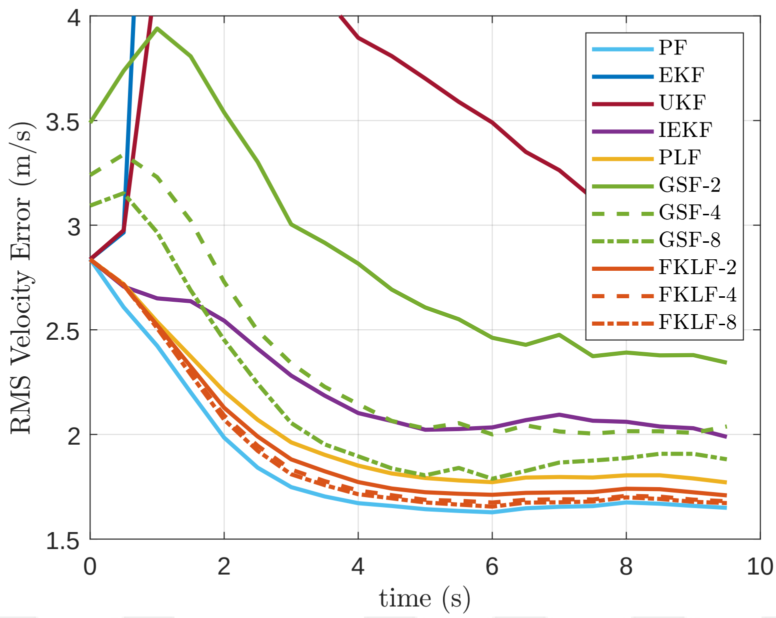



Figure 3.4: Velocity RMSE performance of the filters.

overconfident, i.e., they provide smaller covariances than the true covariance of the estimation error. The overconfidence is especially pronounced for filters providing a Gaussian posterior. GSF can obtain acceptable NEES values only for high number of components. We observe that FKLF results in lower NEES values as the number of components is increased and FKLF-8 asymptotically reaches the NEES values of PF.

The average numbers of iterations required for the convergence of iterative algorithms are illustrated in Figure 3.6. The figure shows that increasing the number of components in FKLF increases the required number of iterations for convergence. The average number of iterations for convergence in FKLF is approximately linear in $J$, the number of components, which makes the computational complexity of FKLF $\mathcal{O}(n_x^3 J^2)$.

### 3.7 Conclusions

A Gaussian mixture/sum filter, named FKLF, has been proposed for the filtering problem with nonlinear measurements. The proposed filter, which is derived by minimizing forward KL divergence approximately, turns out to be a generalization of PLF to

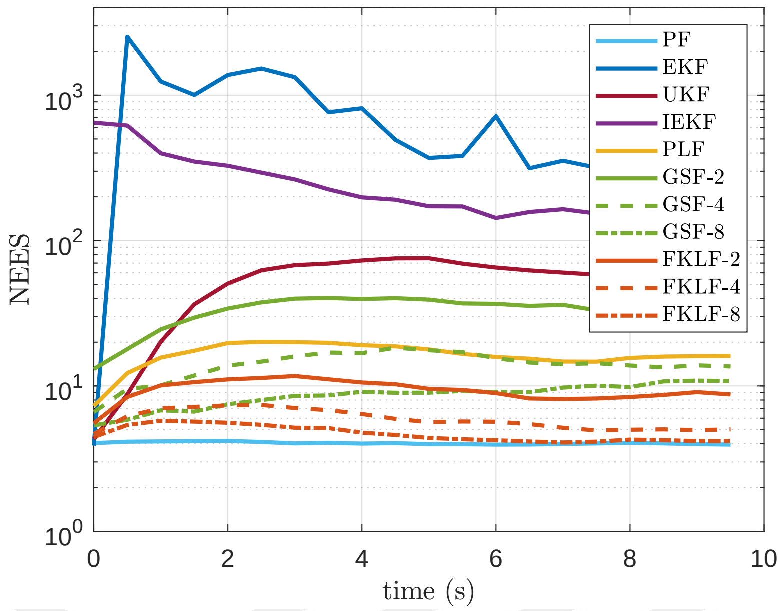



Figure 3.5: NEES values of the filters.

Gaussian mixture posteriors. FKLF fills the gap in the literature between the research on Gaussian sum filtering and nonlinear filtering with divergence minimization. The results show the advantages of FKLF compared to Gaussian filters as well as the standard GSF.

FKLF has so far been tested mostly on scenarios involving unimodal posteriors with non-ellipsoidal support. We believe that the filter can successfully be used in scenarios with multi-modal posteriors if suitable mechanisms are devised to distribute the Gaussian components into different modes.

It is seen in the simulations that FKLF can reach the performance of a PF with moderately low number of components and hence it can be preferable to PF if analytical posterior approximations are desired as opposed to Monte Carlo approximations. It is also our belief that FKLF can be of interest especially for the state estimation of high dimensional systems where PFs suffer from curse of dimensionality.

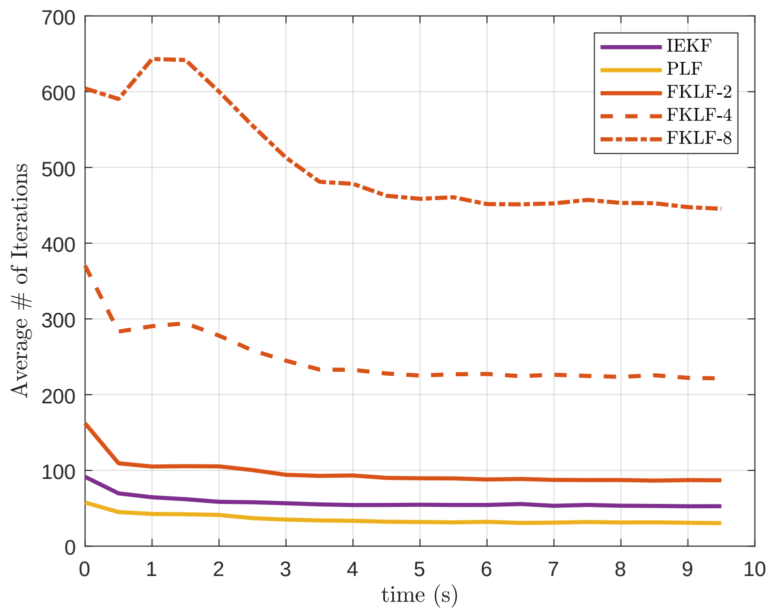



Figure 3.6: Average number of iterations made by the iterative filters.

## 3.8 Derivations

### 3.8.1 Derivations of (3.9) and (3.21)

We provide derivations for the approximate expressions in (3.9) and (3.21) in separate subsections.

#### 3.8.1.1 Derivation of the Gradient $\nabla_{\hat{\mathbf{x}}_i} L(q)$ in (3.9a)

We can write the gradient of the first term on the right hand side of (3.8) with respect to an arbitrary parameter vector $\boldsymbol{\theta}$ as

$$\nabla_{\boldsymbol{\theta}} \mathrm{E}_q[\log q(\mathbf{x})] = \int (\nabla_{\boldsymbol{\theta}} q(\mathbf{x})) \log q(\mathbf{x}) \, \mathrm{d}\mathbf{x}, \tag{3.57}$$

which gives

$$\nabla_{\hat{\mathbf{x}}_i} \mathrm{E}_q[\log q(\mathbf{x})] = \int (\nabla_{\hat{\mathbf{x}}_i} q(\mathbf{x})) \log q(\mathbf{x}) \, \mathrm{d}\mathbf{x} \tag{3.58a}$$

$$= \pi_i \int \nabla_{\hat{\mathbf{x}}_i} q_i(\mathbf{x}) \log q(\mathbf{x}) \, \mathrm{d}\mathbf{x} \tag{3.58b}$$

$$= \pi_i \int q_i(\mathbf{x}) \nabla_{\mathbf{x}} \log q(\mathbf{x}) \, \mathrm{d}\mathbf{x} \tag{3.58c}$$

$$= \pi_i \, \mathrm{E}_{q_i}[\nabla_{\mathbf{x}} \log q(\mathbf{x})], \tag{3.58d}$$

where we used Lemma 1 in Section 3.8.4. We can now approximate the gradient $\nabla_{\mathbf{x}} \log q(\mathbf{x})$ in the expectation as

$$\nabla_{\mathbf{x}} \log q(\mathbf{x}) = -\frac{\sum_{j=1}^{J} \pi_j \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_j, \mathbf{\Sigma}_j) \mathbf{\Sigma}_j^{-1} (\mathbf{x} - \hat{\mathbf{x}}_j)}{\sum_{\ell=1}^{J} \pi_\ell \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_\ell, \mathbf{\Sigma}_\ell)} \tag{3.59a}$$

$$\approx -\sum_{j=1}^{J} \gamma_{i,j} \mathbf{\Sigma}_j^{-1} (\mathbf{x} - \hat{\mathbf{x}}_j), \tag{3.59b}$$

where

$$\gamma_{i,j} \triangleq \frac{\pi_j \mathcal{N}(\hat{\mathbf{x}}_i; \hat{\mathbf{x}}_j, \mathbf{\Sigma}_j)}{\sum_{\ell=1}^{J} \pi_\ell \mathcal{N}(\hat{\mathbf{x}}_i; \hat{\mathbf{x}}_\ell, \mathbf{\Sigma}_\ell)}. \tag{3.60}$$

The approximation above amounts to assuming that

$$\frac{\pi_j q_j(\mathbf{x})}{\sum_{\ell=1}^{J} \pi_\ell q_\ell(\mathbf{x})} q_i(\mathbf{x}) \approx \frac{\pi_j q_j(\hat{\mathbf{x}}_i)}{\sum_{\ell=1}^{J} \pi_\ell q_\ell(\hat{\mathbf{x}}_i)} q_i(\mathbf{x}). \tag{3.61}$$

Substituting (3.59b) into (3.58d), we can write the following approximation for $\nabla_{\hat{\mathbf{x}}_i} \mathrm{E}_q[\log q(\mathbf{x})]$.

$$\nabla_{\hat{\mathbf{x}}_i} \mathrm{E}_q[\log q(\mathbf{x})] \approx -\pi_i \sum_{j=1}^{J} \gamma_{i,j} \mathbf{\Sigma}_j^{-1} (\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j). \tag{3.62}$$

We can calculate the gradient of the second term on the right hand side of (3.8) with respect to $\hat{\mathbf{x}}_i$ as

$$\nabla_{\hat{\mathbf{x}}_i} \mathrm{E}_q[\log p(\mathbf{x})] = \int \nabla_{\hat{\mathbf{x}}_i} q(\mathbf{x}) \log p(\mathbf{x}) \, \mathrm{d}\mathbf{x} \tag{3.63a}$$

$$= \pi_i \int \nabla_{\hat{\mathbf{x}}_i} q_i(\mathbf{x}) \log p(\mathbf{x}) \, \mathrm{d}\mathbf{x} \tag{3.63b}$$

$$= \pi_i \int q_i(\mathbf{x}) \nabla_{\mathbf{x}} \log p(\mathbf{x}) \, \mathrm{d}\mathbf{x} \tag{3.63c}$$

$$= \pi_i \, \mathrm{E}_{q_i}[\nabla_{\mathbf{x}} \log p(\mathbf{x})], \tag{3.63d}$$

where we used Lemma 1 in Section 3.8.4. We can now approximate the gradient $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ in the expectation as

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}) = -\frac{\sum_{n=1}^{N} \pi_{0n} \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{0n}, \mathbf{\Sigma}_{0n}) \mathbf{\Sigma}_{0n}^{-1} (\mathbf{x} - \hat{\mathbf{x}}_{0n})}{\sum_{\ell=1}^{J} \pi_{0\ell} \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{0\ell}, \mathbf{\Sigma}_{0\ell})} \tag{3.64a}$$

$$\approx -\sum_{n=1}^{N} \beta_{i,n} \mathbf{\Sigma}_{0n}^{-1} (\mathbf{x} - \hat{\mathbf{x}}_{0n}), \tag{3.64b}$$

where

$$\beta_{i,n} \triangleq \frac{\pi_{0n} \mathcal{N}(\hat{\mathbf{x}}_i; \hat{\mathbf{x}}_{0n}, \mathbf{\Sigma}_{0n})}{\sum_{\ell=1}^{N} \pi_{0\ell} \mathcal{N}(\hat{\mathbf{x}}_i; \hat{\mathbf{x}}_{0\ell}, \mathbf{\Sigma}_{0\ell})}. \tag{3.65}$$

Note that the approximation above amounts to assuming that

$$\frac{\pi_{0n} p_{0n}(\mathbf{x})}{\sum_{\ell=1}^{J} \pi_{0\ell} p_{0\ell}(\mathbf{x})} q_i(\mathbf{x}) \approx \frac{\pi_{0n} p_{0n}(\hat{\mathbf{x}}_i)}{\sum_{\ell=1}^{J} \pi_{0\ell} p_{0\ell}(\hat{\mathbf{x}}_i)} q_i(\mathbf{x}). \tag{3.66}$$

Substituting (3.64b) into (3.63d), we get

$$\nabla_{\hat{\mathbf{x}}_i} \mathrm{E}_q[\log p(\mathbf{x})] \approx -\pi_i \sum_{n=1}^{N} \beta_{i,n} \mathbf{\Sigma}_{0n}^{-1} (\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_{0n}). \tag{3.67}$$

We can finally easily calculate the gradient of the third term on the right hand side of (3.8) with respect to $\hat{\mathbf{x}}_i$ as

$$\nabla_{\hat{\mathbf{x}}_i} \mathrm{E}_q[\log p(\mathbf{y}|\mathbf{x})] = \pi_i \, \mathrm{E}_{q_i}[\nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x})], \tag{3.68}$$

where we used Lemma 1 in Section 3.8.4. We can now combine the results (3.62), (3.67) and (3.68) to approximate the gradient $\nabla_{\hat{\mathbf{x}}_i} L(q)$ as in (3.9a).

#### 3.8.1.2 Derivation of the Gradient $\nabla_{\mathbf{\Sigma}_i} L(q)$ in (3.9b)

We can write the gradient of the first term on the right hand side of (3.8) with respect to $\mathbf{\Sigma}_i$ as

$$\begin{aligned}
\nabla_{\mathbf{\Sigma}_i} \mathrm{E}_q[\log q(\mathbf{x})] &= \int \nabla_{\mathbf{\Sigma}_i} q(\mathbf{x}) \log q(\mathbf{x}) \, \mathrm{d}\mathbf{x} &\text{(3.69a)}\\
&= \pi_i \int \nabla_{\mathbf{\Sigma}_i} q_i(\mathbf{x}) \log q(\mathbf{x}) \, \mathrm{d}\mathbf{x} &\text{(3.69b)}\\
&= \frac{1}{2} \pi_i \int q_i(\mathbf{x}) \nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^{\mathrm{T}} \log q(\mathbf{x}) \, \mathrm{d}\mathbf{x} &\text{(3.69c)}\\
&= \frac{1}{2} \pi_i \, \mathrm{E}_{q_i}[\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^{\mathrm{T}} \log q(\mathbf{x})], &\text{(3.69d)}
\end{aligned}$$

where we used Lemma 2 in Section 3.8.4. We can now approximate the Hessian $\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^{\mathrm{T}} \log q(\mathbf{x})$ in the expectation as

$$\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^{\mathrm{T}} \log q(\mathbf{x}) \approx -\sum_{j=1}^{J} \gamma_{i,j} \mathbf{\Sigma}_j^{-1}, \tag{3.70}$$

which was written by taking the gradient of the approximation in (3.59b) with respect to $\mathbf{x}$. Substituting (3.70) into (3.69d), we get

$$\nabla_{\boldsymbol{\Sigma}_i} \mathrm{E}_q[\log q(\mathbf{x})] \approx -\frac{1}{2}\pi_i \sum_{j=1}^{J} \gamma_{i,j} \boldsymbol{\Sigma}_j^{-1}. \tag{3.71}$$

We can write the gradient of the second term on the right hand side of (3.8) with respect to $\boldsymbol{\Sigma}_i$ as

$$\begin{aligned}
\nabla_{\boldsymbol{\Sigma}_i} \mathrm{E}_q[\log p(\mathbf{x})] &= \int \nabla_{\boldsymbol{\Sigma}_i} q(\mathbf{x}) \log p(\mathbf{x}) \, \mathrm{d}\mathbf{x} &(3.72\text{a})\\
&= \pi_i \int \nabla_{\boldsymbol{\Sigma}_i} q_i(\mathbf{x}) \log p(\mathbf{x}) \, \mathrm{d}\mathbf{x} &(3.72\text{b})\\
&= \frac{1}{2}\pi_i \int q_i(\mathbf{x}) \nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^{\mathrm{T}} \log p(\mathbf{x}) \, \mathrm{d}\mathbf{x} &(3.72\text{c})\\
&= \frac{1}{2}\pi_i \, \mathrm{E}_{q_i}[\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^{\mathrm{T}} \log p(\mathbf{x})], &(3.72\text{d})
\end{aligned}$$

where we used Lemma 2 in Section 3.8.4. We can approximate the Hessian $\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^{\mathrm{T}} \log p(\mathbf{x})$ in the expectation as

$$\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^{\mathrm{T}} \log p(\mathbf{x}) \approx -\sum_{n=1}^{N} \beta_{i,n} \boldsymbol{\Sigma}_{0n}^{-1}, \tag{3.73}$$

which was written by taking the gradient of the approximation in (3.64b) with respect to $\mathbf{x}$. Substituting (3.73) into (3.72d), we get

$$\nabla_{\boldsymbol{\Sigma}_i} \mathrm{E}_q[\log p(\mathbf{x})] \approx -\frac{1}{2}\pi_i \sum_{n=1}^{N} \beta_{i,n} \boldsymbol{\Sigma}_{0n}^{-1}. \tag{3.74}$$

We can finally easily calculate the gradient of the third term on the right hand side of (3.8) with respect to $\boldsymbol{\Sigma}_i$ as

$$\nabla_{\boldsymbol{\Sigma}_i} \mathrm{E}_q[\log p(\mathbf{y}|\mathbf{x})] = \frac{1}{2}\pi_i \, \mathrm{E}_{q_i}[\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^{\mathrm{T}} \log p(\mathbf{y}|\mathbf{x})], \tag{3.75}$$

where we used Lemma 2 in Section 3.8.4. We can now combine the results (3.71), (3.74) and (3.75) to approximate the gradient $\nabla_{\boldsymbol{\Sigma}_i} L(q)$ as in (3.9b).

#### 3.8.1.3 Derivation of the Hessian $\nabla_{\hat{\mathbf{x}}_i}\nabla^{\mathrm{T}}_{\hat{\mathbf{x}}_i}L(q)$ in (3.21)

Taking the gradient $\nabla_{\hat{\mathbf{x}}_i}L(q)$ in (3.9a) with respect to $\hat{\mathbf{x}}_i$, we can approximate the Hessian $\nabla_{\hat{\mathbf{x}}_i}\nabla^{\mathrm{T}}_{\hat{\mathbf{x}}_i}L(q)$ as

$$\begin{aligned}\nabla_{\hat{\mathbf{x}}_i}\nabla^{\mathrm{T}}_{\hat{\mathbf{x}}_i}L(q) \approx &- \pi_i \sum_{j\notin\mathcal{N}_i}\gamma_{i,j}\boldsymbol{\Sigma}_j^{-1} + \pi_i\sum_{n=1}^{N}\beta_{i,n}\boldsymbol{\Sigma}_{0n}^{-1} \\ &- \pi_i\,\mathrm{E}_{q_i}\left[\nabla_{\mathbf{x}}\nabla^{\mathrm{T}}_{\mathbf{x}}\log p(\mathbf{y}|\mathbf{x})\right], \end{aligned} \tag{3.76}$$

where we used Lemma 1 in Section 3.8.4 and we assumed that $\nabla_{\hat{\mathbf{x}}_i}\gamma_{i,j} \approx \mathbf{0}$ and $\nabla_{\hat{\mathbf{x}}_i}\beta_{i,n} \approx \mathbf{0}$. Using (3.9b) on the right hand side of (3.76), we get

$$\nabla_{\hat{\mathbf{x}}_i}\nabla^{\mathrm{T}}_{\hat{\mathbf{x}}_i}L(q) \approx 2\nabla_{\boldsymbol{\Sigma}_i}L(q) + \pi_i\sum_{j\in\mathcal{N}_i}\gamma_{i,j}\boldsymbol{\Sigma}_j^{-1} \tag{3.77a}$$

$$= 2\nabla_{\boldsymbol{\Sigma}_i}L(q) + \pi_i\boldsymbol{\Sigma}_i^{-1}\sum_{j\in\mathcal{N}_i}\gamma_{i,j} \tag{3.77b}$$

$$\approx 2\nabla_{\boldsymbol{\Sigma}_i}L(q) + \pi_i\boldsymbol{\Sigma}_i^{-1}, \tag{3.77c}$$

where we used the approximation $\sum_{j\in\mathcal{N}_i}\gamma_{i,j} \approx 1$, which is justified in Remark 3. Assuming also that the covariance $\boldsymbol{\Sigma}_i$ is close to its optimal value would mean that we have $\nabla_{\boldsymbol{\Sigma}_i}L(q) \approx \mathbf{0}$, which gives the Hessian approximation in (3.21).

### 3.8.2 Derivation of (3.24a)

The first term on the right hand side of (3.23b) is given as follows.

$$\mathrm{E}_q[\log q(\mathbf{x})] \triangleq \sum_{j=1}^{J}\pi_j E_{q_j}[\log q(\mathbf{x})]. \tag{3.78a}$$

Since the expectation $\mathrm{E}_{q_j}[\log q(\mathbf{x})]$ above is analytically intractable, we are going to replace it with its best variational upper bound as follows. Using the concavity property of logarithm, we can bound $\log q(\mathbf{x})$ by its first order Taylor series expansion around an arbitrary positive constant $\bar{q}_j > 0$ to write

$$\mathrm{E}_{q_j}[\log q(\mathbf{x})] \leq \mathrm{E}_{q_j}\left[\log\bar{q}_j + \frac{q(\mathbf{x}) - \bar{q}_j}{\bar{q}_j}\right] \tag{3.79a}$$

$$= \log\bar{q}_j + \frac{\mathrm{E}_{q_j}[q(\mathbf{x})]}{\bar{q}_j} - 1. \tag{3.79b}$$

We can minimize the upper bound above with respect to $\bar{q}_j > 0$ to find

$$\bar{q}_j^* = \mathrm{E}_{q_j}[q(\mathbf{x})]. \tag{3.80}$$

Substituting $\bar{q}_j^*$ into the right hand side of (3.79b) gives the following approximation of $\mathrm{E}_{q_j}[\log q(\mathbf{x})]$.

$$\mathrm{E}_{q_j}[\log q(\mathbf{x})] \approx \log \mathrm{E}_{q_j}[q(\mathbf{x})] \tag{3.81a}$$

$$= \log \left( \sum_{\ell=1}^{J} \pi_\ell \mathcal{N}(\hat{\mathbf{x}}_j; \hat{\mathbf{x}}_\ell, \mathbf{\Sigma}_j + \mathbf{\Sigma}_\ell) \right). \tag{3.81b}$$

Based on the approximation above, we can write the final expression for $\mathrm{E}_q[\log q(\mathbf{x})]$ as follows.

$$\mathrm{E}_q[\log q(\mathbf{x})] = \sum_{j=1}^{J} \pi_j \log \left( \sum_{\ell=1}^{J} \pi_\ell \mathcal{N}(\hat{\mathbf{x}}_j; \hat{\mathbf{x}}_\ell, \mathbf{\Sigma}_j + \mathbf{\Sigma}_\ell) \right). \tag{3.82}$$

The second term on the right hand side of (3.23b) is given as

$$\mathrm{E}_q[\log p(\mathbf{x})] = \sum_{j=1}^{J} \pi_j \, \mathrm{E}_{q_j}[\log p(\mathbf{x})]. \tag{3.83}$$

We can again replace the analytically intractable expectation $\mathrm{E}_{q_j}[\log p(\mathbf{x})]$ with its best variational upper bound, as was done above for $\mathrm{E}_{q_j}[\log q(\mathbf{x})]$, to end up with the final expression for $\mathrm{E}_q[\log p(\mathbf{x})]$ as follows.

$$\mathrm{E}_q[\log p(\mathbf{x})] = \sum_{j=1}^{J} \pi_j \log \left( \sum_{n=1}^{N} \pi_{0n} \mathcal{N}(\hat{\mathbf{x}}_j; \hat{\mathbf{x}}_{0n}, \mathbf{\Sigma}_j + \mathbf{\Sigma}_{0n}) \right). \tag{3.84}$$

We can now obtain (3.24a) by combining the results (3.82) and (3.84).

### 3.8.3 Derivation of (3.56)

In this subsection, we provide the derivation of equation (3.56). The form of the Gaussian mixture prior is provided in equation (3.55). We can write the proposed Gaussian mixture prior as

$$p(\mathbf{x}_0) = \sum_{j=1}^{J} \frac{1}{J} \mathcal{N}(\mathbf{x}_0; \bar{\mathbf{x}}_0^j, \mathbf{\Sigma}), \tag{3.85}$$

where $\bar{\mathbf{x}}_0^j \sim \mathcal{N}(\bar{\mathbf{x}}_0^j; \bar{\mathbf{x}}_0, \mathbf{\Sigma})$. This means that we select the means of the mixture's components at random. Furthermore, the covariances of the Gaussian mixture components are set to be identical. We first show that the expected mean of $\mathbf{x}_0$ is equal to $\bar{\mathbf{x}}_0$.

$$\mathrm{E}\left[\mathrm{E}[\mathbf{x}_0]\right] = \mathrm{E}\left[\sum_{j=1}^{J} \frac{1}{J} \bar{\mathbf{x}}_0^j\right] \tag{3.86a}$$

$$= \frac{1}{J} \sum_{j=1}^{J} \bar{\mathbf{x}}_0 \tag{3.86b}$$

$$= \bar{\mathbf{x}}_0. \tag{3.86c}$$

Note that we first take expectation with respect to $\mathbf{x}_0$, then $\bar{\mathbf{x}}_0^j$ for $j = 1, \ldots, J$. Now, we need to determine the covariance $\mathbf{\Sigma}$ such that the expected covariance of the random mixture $p(\mathbf{x}_0)$ becomes $\mathbf{\Sigma}_0$. We can write the covariance of $p(\mathbf{x}_0)$ as

$$\mathrm{Cov}(\mathbf{x}_0) = \sum_{j=1}^{J} \frac{1}{J} \left(\mathbf{\Sigma} + \bar{\mathbf{x}}_0^j \bar{\mathbf{x}}_0^{j,\mathrm{T}}\right) - \left(\sum_{j=1}^{J} \frac{1}{J} \bar{\mathbf{x}}_0^j\right) \left(\sum_{j=1}^{J} \frac{1}{J} \bar{\mathbf{x}}_0^j\right)^{\mathrm{T}} \tag{3.87a}$$

$$= \mathbf{\Sigma} + \frac{1}{J} \sum_{j=1}^{J} \bar{\mathbf{x}}_0^j \bar{\mathbf{x}}_0^{j,\mathrm{T}} - \left(\frac{1}{J} \sum_{j=1}^{J} \bar{\mathbf{x}}_0^j\right) \left(\frac{1}{J} \sum_{j=1}^{J} \bar{\mathbf{x}}_0^j\right)^{\mathrm{T}}, \tag{3.87b}$$

where $\bar{\mathbf{x}}_0^{j,\mathrm{T}}$ represents the transpose of $\bar{\mathbf{x}}_0^j$. We take the expectation of this covariance expression as

$$\mathrm{E}[\mathrm{Cov}(\mathbf{x}_0)] = \mathbf{\Sigma} + \frac{1}{J} \sum_{j=1}^{J} \mathrm{E}\left[\bar{\mathbf{x}}_0^j \bar{\mathbf{x}}_0^{j,\mathrm{T}}\right] - \mathrm{E}\left[\left(\frac{1}{J} \sum_{j=1}^{J} \bar{\mathbf{x}}_0^j\right) \left(\frac{1}{J} \sum_{j=1}^{J} \bar{\mathbf{x}}_0^j\right)^{\mathrm{T}}\right] \tag{3.88a}$$

$$= \mathbf{\Sigma} + \frac{1}{J} \sum_{j=1}^{J} \left(\mathbf{\Sigma} + \bar{\mathbf{x}}_0 \bar{\mathbf{x}}_0^{\mathrm{T}}\right) - \mathrm{E}\left[\frac{1}{J} \sum_{j=1}^{J} \bar{\mathbf{x}}_0^j\right] \mathrm{E}\left[\frac{1}{J} \sum_{j=1}^{J} \bar{\mathbf{x}}_0^{j,\mathrm{T}}\right] - \mathrm{Cov}\left(\frac{1}{J} \sum_{j=1}^{J} \bar{\mathbf{x}}_0^j\right) \tag{3.88b}$$

$$= \frac{2J-1}{J} \mathbf{\Sigma}. \tag{3.88c}$$

We can equate expected covariance of $p(\mathbf{x}_0)$ to $\mathbf{\Sigma}_0$ and find the covariance $\mathbf{\Sigma}$ as

$$\mathbf{\Sigma} = \frac{J}{2J-1} \mathbf{\Sigma}_0 \tag{3.89}$$

### 3.8.4 Lemmas Used in the Derivations

In this part, we present the lemmas utilized in the previous subsections' derivations.

**Lemma 1.** *Let* $q(\mathbf{x})$ *denote the Gaussian distribution* $\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{\Sigma})$. *The following identity holds.*

$$\int (\nabla_{\hat{\mathbf{x}}} q(\mathbf{x}))\, f(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{\Sigma})\; \mathrm{d}\mathbf{x} = \mathrm{E}_q[\nabla_{\mathbf{x}} f(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{\Sigma})], \tag{3.90}$$

*where* $f(\cdot, \cdot)$ *is an arbitrary continuously differentiable function of* $\mathbf{x}$, $\hat{\mathbf{x}}$ *and* $\mathbf{\Sigma}$ *satisfying* $\lim_{\|\mathbf{x}\| \to \infty} f(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{\Sigma}) q(\mathbf{x}) = 0$ *for all* $\hat{\mathbf{x}}$ *and* $\mathbf{\Sigma}$.

*Proof.* Since we have $\nabla_{\hat{\mathbf{x}}} q(\mathbf{x}) = -\nabla_{\mathbf{x}} q(\mathbf{x})$, we can write the left hand side of (3.90) as

$$\int (\nabla_{\hat{\mathbf{x}}} q(\mathbf{x}))\, f(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{\Sigma})\; \mathrm{d}\mathbf{x} = -\int (\nabla_{\mathbf{x}} q(\mathbf{x}))\, f(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{\Sigma})\; \mathrm{d}\mathbf{x}. \tag{3.91}$$

We can now apply integration by parts on the right hand side by choosing $\mathbf{u} \triangleq f(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{\Sigma})$ and $\mathrm{d}\mathbf{v} \triangleq \nabla_{\mathbf{x}} q(\mathbf{x})$ to get

$$\int (\nabla_{\hat{\mathbf{x}}} q(\mathbf{x}))\, f(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{\Sigma})\; \mathrm{d}\mathbf{x} = \int q(\mathbf{x}) \nabla_{\mathbf{x}} f(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{\Sigma})\; \mathrm{d}\mathbf{x} \tag{3.92a}$$

$$= \mathrm{E}_q[\nabla_{\mathbf{x}} f(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{\Sigma})], \tag{3.92b}$$

which completes the proof. □

**Lemma 2.** *Let* $q(\mathbf{x})$ *denote the Gaussian distribution* $\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{\Sigma})$. *The following identity holds.*

$$\int (\nabla_{\mathbf{\Sigma}} q(\mathbf{x}))\, f(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{\Sigma})\; \mathrm{d}\mathbf{x} = \frac{1}{2}\, \mathrm{E}_q[\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^{\mathrm{T}} f(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{\Sigma})] \tag{3.93}$$

*where* $f(\cdot, \cdot, \cdot)$ *is an arbitrary continuously differentiable function of* $\mathbf{x}$, $\hat{\mathbf{x}}$, *and* $\mathbf{\Sigma}$ *satisfying* $\lim_{\|\mathbf{x}\| \to \infty} q(\mathbf{x}) \nabla_{\mathbf{x}} f(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{\Sigma}) = \mathbf{0}$ *and* $\lim_{\|\mathbf{x}\| \to \infty} f(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{\Sigma}) \nabla_{\mathbf{x}} q(\mathbf{x}) = \mathbf{0}$ *for all* $\hat{\mathbf{x}}$ *and* $\mathbf{\Sigma}$.

*Proof.* Since we have $\nabla_{\mathbf{\Sigma}} q(\mathbf{x}) = \frac{1}{2} \nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^{\mathrm{T}} q(\mathbf{x})$, we can write the left hand side of (3.93) as

$$\int (\nabla_{\mathbf{\Sigma}} q(\mathbf{x})) f(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{\Sigma})\; \mathrm{d}\mathbf{x} = \frac{1}{2} \int (\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^{\mathrm{T}} q(\mathbf{x})) f(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{\Sigma})\; \mathrm{d}\mathbf{x}. \tag{3.94}$$

The rest of the proof follows by applying integration by parts twice on the integral on the right hand side above. □

Note that although Lemma 1 and Lemma 2 resemble Bonnet [53] and Price [54] identities, respectively [55, Appendix B], they are different since in Bonnet and Price identities, the functions $f(\cdot)$ do not depend on mean and covariance.

# CHAPTER 4

# BAYESIAN FILTERING/SMOOTHING WITH UNKNOWN NOISE COVARIANCES

In this chapter, Bayesian filtering and smoothing problems with unknown process and measurement noise covariances are investigated for linear Gaussian systems, separately. The process and measurement noise covariances are assumed to be inverse Wishart distributed. Bayesian filters and smoothers calculating the joint posteriors for the state and the unknown noise covariances are derived by using a scale Gaussian approximation of t-distribution and moment matching. The proposed filters and smoothers are non-iterative unlike the existing Bayesian solutions in the literature. The performance of the proposed algorithms is illustrated on a two-dimensional target tracking scenario. The simulation results show that the proposed filters and smoothers have similar performance as the state of the art solutions with lower computational load.

## 4.1 Introduction

Kalman filter and smoother are the optimal state estimators for linear Gaussian systems [56] in the minimum mean square error (MMSE) sense when all of the model parameters are perfectly known. On the other hand, in many real-life applications, there exist unknown or uncertain model parameters which might affect the estimation performance. One such example is the measurement noise covariance (MNC), which quantifies the imperfections of the sensor. The classical Kalman filter/smoother literature assumes that the MNC is perfectly known. However, in applications like radar target tracking, the MNC might depend on several factors like environmental conditions, clutter properties, target SNR etc. which makes it difficult or impossible to know the exact value of the covariance. Furthermore, since these factors can be time/space varying, the true MNC can be time/space varying. Another example of unknown or uncertain model parameters for Kalman filter/smoother can be process noise covariance (PNC), which represents uncertainties in the state model. In applications like radar target tracking, the PNC models the target maneuvers, which are impossible to know. When the tracker uses a wrong MNC or PNC, the consequences can range from minor performance degradation to filter divergence [57, 58]. A solution to avoid these consequences is to estimate the unknown covariances together with the states, which is main topic of this chapter.

The following categorization can be made about the literature related to state estimation with unknown noise covariances.

- **Non-random noise covariances estimation:** In this literature, the unknown noise covariances are modeled as non-random parameters:

  In [59, 60], the authors formulate the problem of smoothing with unknown noise covariances by using maximum likelihood method. Innovation based approaches are used in [61–65]. Correlation, covariance matching and maximum likelihood based methods are investigated in [66]. In [67], noise scale is estimated instead of the full noise covariances based on Kullback-Leibler divergence minimization.

- **Bayesian (random) noise covariances estimation:** In this literature the unknown noise covariances are modeled as random parameters and Bayesian estimation is used. The Bayesian noise covariance estimation methods can be divided into two classes: implicit (marginalization based) and explicit Bayesian noise covariance estimation approaches.

  – **Implicit (marginalization based) Bayesian noise covariance estimation:** In these methods the unknown noise covariances are marginalized out from the problem. This approach is also equivalent to replacing the Gaussian noises with unknown noise covariances into Student's *t* distributed noises and also known as state estimation under heavy-tailed noises [66, 68–71]. In implicit approaches, the main aim is to come up with state estimation algorithms which are robust to unknown noise covariances and no explicit estimates or posteriors for the noise covariances are calculated, i.e., noise covariances are only implicitly estimated.

- **Explicit Bayesian noise covariance estimation:** In these methods, marginalization is not used and explicit estimates can be calculated for the unknown noise covariances. In [72] and [73], variational Bayes (VB) approach [37, 74] is used and the problem is solved by using variational Bayes expectation maximization and fixed point iteration techniques, respectively. The authors in [73] assume that the PNC is known and the unknown MNC is diagonal with inverse gamma distributed entries. The work in [73] is extended to jump Markov linear and nonlinear systems in [75] and [76], respectively. Moreover, the work in [73] has been improved in [77] to estimate non-diagonal noise covariances for nonlinear systems. In [78–84], inverse Wishart (IW) distribution is assumed on the noise covariances and the problem is solved using VB technique. Some other works dealing with unknown noise statistics and solving the problem using VB are given in [85–87]. In [88], noise covariances are found recursively by using maximum a posteriori (MAP) estimation. The authors in [89] investigate adaptive state estimation with unknown process noise statistics by using a modified version of the algorithm in [88], which introduces a forgetting factor between previous and current PNC estimates. In [90], noise adaptive filtering is studied using approximations and the properties of the Student's $t$ and IW distributions. The filter utilizes a fixed point approach to find the filtered distribution of the state and the estimates of noise covariances used in the filter.

In this chapter, we propose explicit Bayesian algorithms for linear Gaussian filtering and smoothing problems with unknown PNC and MNC, separately. The joint posteriors of the states and the noise covariances are calculated by utilizing a $t$-distribution approximation based on its the scale mixture property and moment-matching approximations. Although most of the algorithms in the literature require iterations to converge to a posterior, the proposed algorithms are non-iterative, which makes them rather efficient.

The remainder of the chapter is organized as follows: In Section 4.2, the problem definition is provided. Filtering with unknown process and measurement covariances are investigated in Sections 4.3 and 4.4, respectively. Corresponding smoothing solutions for unknown process and measurement noise covariances are given in Sections 4.5 and 4.6, respectively. In Section 4.7, numerical examples that illustrate the obtained results are presented. In Section 4.8, the case where both noise covariances are unknown is investigated. Conclusions are presented in Section 4.9. Finally, the derivations are provided in Section 4.10.

## 4.2 Problem Definition

We consider the linear Gaussian system

$$\mathbf{x}_{n+1} = \mathbf{A}\mathbf{x}_n + \boldsymbol{\omega}_{n+1}, \qquad \boldsymbol{\omega}_{n+1} \sim \mathcal{N}(\boldsymbol{\omega}_{n+1}; \mathbf{0}, \mathbf{Q}_{n+1}), \tag{4.1a}$$

$$\mathbf{y}_n = \mathbf{C}\mathbf{x}_n + \mathbf{v}_n, \qquad \mathbf{v}_n \sim \mathcal{N}(\mathbf{v}_n; \mathbf{0}, \mathbf{R}_n), \tag{4.1b}$$

for $n = 0, \dots, N$ where $\mathbf{x}_n \in \mathbb{R}^{n_x}$ shows the state vector of dimension $n_x$ and $\mathbf{y}_n \in \mathbb{R}^{n_y}$ is the $n_y$-dimensional measurement. The matrices $\mathbf{A}$ and $\mathbf{C}$ are state transition and measurement matrices, respectively. $\boldsymbol{\omega}_n$ and $\mathbf{v}_n$ show the Gaussian distributed zero-mean and white process and measurement noises, respectively. The time-dependent PNC $\mathbf{Q}_n$ and MNC $\mathbf{R}_n$ are unknown and random with models

$$\mathbf{Q}_{n+1}|\mathbf{Q}_n \sim p(\mathbf{Q}_{n+1}|\mathbf{Q}_n), \qquad \mathbf{Q}_0 \sim p(\mathbf{Q}_0), \tag{4.2a}$$

$$\mathbf{R}_{n+1}|\mathbf{R}_n \sim p(\mathbf{R}_{n+1}|\mathbf{R}_n), \qquad \mathbf{R}_0 \sim p(\mathbf{R}_0), \tag{4.2b}$$

where $p(\mathbf{Q}_{n+1}|\mathbf{Q}_n)$ and $p(\mathbf{R}_{n+1}|\mathbf{R}_n)$ denote the beta-Bartlett transition model for inverse Wishart (IW) distributed PNC and MNC [79] with discount factors $\gamma_Q$ and $\gamma_R$, respectively. Moreover, we have $p(\mathbf{Q}_0) = \mathcal{IW}(\mathbf{Q}_0; v_0, \mathbf{V}_0)$ and $p(\mathbf{R}_0) = \mathcal{IW}(\mathbf{R}_0; w_0, \mathbf{W}_0)$ where IW distribution is given as [6]

$$\mathcal{IW}(\mathbf{X}; v, \mathbf{V}) = \frac{|\mathbf{V}|^{\frac{1}{2}(v-d-1)} \operatorname{etr}\left(-\frac{1}{2}\mathbf{X}^{-1}\mathbf{V}\right)}{2^{\frac{1}{2}(v-d-1)d}\Gamma_d\left[\frac{1}{2}(v-d-1)\right] |\mathbf{X}|^{\frac{v}{2}}} \tag{4.3}$$

for a random matrix $\mathbf{X} \in \mathbb{R}^{d\times d}$. In this definition, $v$ and $\mathbf{V}$ represent degrees of freedom and scale matrix, respectively; the operation $\operatorname{etr}$ represents the exponential trace function; $\Gamma_d$ is the multivariate gamma function.

The aim of this chapter is to calculate/approximate the filtered posteriors $p(\mathbf{x}_n, \mathbf{Q}_n|\mathbf{y}_{0:n})$, $p(\mathbf{x}_n, \mathbf{R}_n|\mathbf{y}_{0:n})$ and smoothed posteriors $p(\mathbf{x}_n, \mathbf{Q}_n|\mathbf{y}_{0:N})$, $p(\mathbf{x}_n, \mathbf{R}_n|\mathbf{y}_{0:N})$ for $n = 0, \ldots, N$. These distributions are analytically intractable and hence approximations are necessary. We consider the filtering and smoothing problems in the subsequent sections separately.

## 4.3 Bayesian Filtering with Unknown PNC

In this part, we provide the details of Bayesian filtering with an unknown PNC when the MNCs are known. We approximate the filtered posterior $p(\mathbf{x}_n, \mathbf{Q}_n|\mathbf{y}_{0:n})$ as

$$p(\mathbf{x}_n, \mathbf{Q}_n|\mathbf{y}_{0:n}) \approx \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n})\mathcal{IW}(\mathbf{Q}_n; \upsilon_{n|n}, \mathbf{V}_{n|n}). \tag{4.4}$$

Now, we assume the following joint transition density.

$$p(\mathbf{x}_n, \mathbf{Q}_n|\mathbf{x}_{n-1}, \mathbf{Q}_{n-1}) = \mathcal{N}(\mathbf{x}_n; \mathbf{A}\mathbf{x}_{n-1}, \mathbf{Q}_n)p(\mathbf{Q}_n|\mathbf{Q}_{n-1}), \tag{4.5}$$

We can use Chapman-Kolmogorov equation [12] to calculate the distribution $p(\mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{Q}_n|\mathbf{y}_{0:n-1})$ as

$$p(\mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{Q}_n|\mathbf{y}_{0:n-1}) = \int p(\mathbf{x}_n, \mathbf{Q}_n|\mathbf{x}_{n-1}, \mathbf{Q}_{n-1})p(\mathbf{x}_{n-1}, \mathbf{Q}_{n-1}|\mathbf{y}_{0:n-1})\, \mathrm{d}\mathbf{Q}_{n-1} \tag{4.6a}$$

$$\begin{aligned} &= \int \mathcal{N}(\mathbf{x}_n; \mathbf{A}\mathbf{x}_{n-1}, \mathbf{Q}_n)p(\mathbf{Q}_n|\mathbf{Q}_{n-1}) \\ &\quad \times \mathcal{N}(\mathbf{x}_{n-1}; \hat{\mathbf{x}}_{n-1|n-1}, \mathbf{P}_{n-1|n-1})\mathcal{IW}(\mathbf{Q}_{n-1}; \upsilon_{n-1|n-1}, \mathbf{V}_{n-1|n-1})\, \mathrm{d}\mathbf{Q}_{n-1} \qquad (4.6b) \\ &= \mathcal{N}(\mathbf{x}_n; \mathbf{A}\mathbf{x}_{n-1}, \mathbf{Q}_n)\mathcal{N}(\mathbf{x}_{n-1}, \hat{\mathbf{x}}_{n-1|n-1}, \mathbf{P}_{n-1|n-1})\mathcal{IW}(\mathbf{Q}_n; \upsilon_{n|n-1}, \mathbf{V}_{n|n-1}), \qquad (4.6c) \end{aligned}$$

where

$$\upsilon_{n|n-1} = \gamma_Q \upsilon_{n-1|n-1} + (1-\gamma_Q)(2n_x + 2), \qquad \mathbf{V}_{n|n-1} = \gamma_Q \mathbf{V}_{n-1|n-1}. \tag{4.7}$$

The updated posterior distribution $p(\mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{Q}_n|\mathbf{y}_{0:n})$ is found using Bayes' rule as follows:

$$\begin{aligned} p(\mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{Q}_n|\mathbf{y}_{0:n}) &\propto p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{Q}_n|\mathbf{y}_{0:n-1}) \qquad (4.8a) \\ &= \mathcal{N}(\mathbf{y}_n; \mathbf{C}\mathbf{x}_n, \mathbf{R}_n)\mathcal{N}(\mathbf{x}_{n-1}, \hat{\mathbf{x}}_{n-1|n-1}, \mathbf{P}_{n-1|n-1}) \end{aligned}$$

$$\times \mathcal{N}(\mathbf{x}_n; \mathbf{A}\mathbf{x}_{n-1}, \mathbf{Q}_n)\mathcal{IW}(\mathbf{Q}_n; \upsilon_{n|n-1}, \mathbf{V}_{n|n-1}). \tag{4.8b}$$

As can be seen, the term $\mathbf{Q}_n$ exists in both Gaussian and inverse Wishart terms. Due to these terms, we cannot derive the posterior distributions of the state $\mathbf{x}_n$ and covariance $\mathbf{Q}_n$ independently by marginalizing the joint posterior distribution. To solve this problem, we write the multiplication of the terms $\mathcal{N}(\mathbf{x}_n; \mathbf{A}\mathbf{x}_{n-1}, \mathbf{Q}_n)$ and $\mathcal{IW}(\mathbf{Q}_n; \upsilon_{n|n-1}, \mathbf{V}_{n|n-1})$ as the multiplication of Student's *t* and IW distributions as follows.

$$\begin{aligned}\mathcal{N}(\mathbf{x}_n;&\mathbf{A}\mathbf{x}_{n-1}, \mathbf{Q}_n)\mathcal{IW}(\mathbf{Q}_n; \upsilon_{n|n-1}, \mathbf{V}_{n|n-1}) \propto \\ &\text{St}\left(\mathbf{x}_n; \mathbf{A}\mathbf{x}_{n-1}, \frac{\mathbf{V}_{n|n-1}}{\upsilon_{n|n-1} - 2n_x}, \upsilon_{n|n-1} - 2n_x\right) \\ &\times \mathcal{IW}(\mathbf{Q}_n; \upsilon_{n|n-1} + 1, \mathbf{V}_{n|n-1} + (\mathbf{x}_n - \mathbf{A}\mathbf{x}_{n-1})(\cdot)^{\mathrm{T}}),\end{aligned} \tag{4.9}$$

which is proven in Section 4.10.1. The distribution $p(\mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{Q}_n|\mathbf{y}_{0:n})$ can now be written by using (4.9) as

$$\begin{aligned}&p(\mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{Q}_n|\mathbf{y}_{0:n}) \propto \mathcal{N}\left(\mathbf{y}_n; \mathbf{C}\mathbf{x}_n, \mathbf{R}_n\right)\mathcal{N}(\mathbf{x}_{n-1}, \hat{\mathbf{x}}_{n-1|n-1}, \mathbf{P}_{n-1|n-1}) \\ &\times \text{St}\left(\mathbf{x}_n; \mathbf{A}\mathbf{x}_{n-1}, \frac{\mathbf{V}_{n|n-1}}{\upsilon_{n|n-1} - 2n_x}, \upsilon_{n|n-1} - 2n_x\right) \\ &\times \mathcal{IW}(\mathbf{Q}_n; \upsilon_{n|n-1} + 1, \mathbf{V}_{n|n-1} + (\mathbf{x}_n - \mathbf{A}\mathbf{x}_{n-1})(\cdot)^{\mathrm{T}}),\end{aligned} \tag{4.10}$$

where the covariance $\mathbf{Q}_n$ only appears in IW distribution, as desired. It is well known that we can write *t*-distribution as a scale Gaussian mixture [91].

$$\text{St}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu) = \int \mathcal{N}\left(\mathbf{x}, \boldsymbol{\mu}, \frac{\boldsymbol{\Sigma}}{\lambda}\right) \mathcal{G}\left(\lambda; \frac{\nu}{2}, \frac{\nu}{2}\right) \, \mathrm{d}\lambda, \tag{4.11}$$

where $\mathcal{G}(\lambda; \alpha, \beta)$ denotes Gamma distribution with a shape and rate parameter $\alpha$ and $\beta$, respectively. Based on this result, we can rewrite the joint posterior distribution (4.10) by approximating the *t*-distribution with a Gaussian mixture as follows

$$\begin{aligned}p(\mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{Q}_n|\mathbf{y}_{0:n}) &\propto \sum_{j=1}^{J} \alpha_j \mathcal{N}\left(\mathbf{y}_n; \mathbf{C}\mathbf{x}_n, \mathbf{R}_n\right) \\ &\times \mathcal{N}(\mathbf{x}_{n-1}; \hat{\mathbf{x}}_{n-1|n-1}, \mathbf{P}_{n-1|n-1})\mathcal{N}\left(\mathbf{x}_n; \mathbf{A}\mathbf{x}_{n-1}, \frac{\overline{\mathbf{Q}}_{n|n-1}}{u_j}\right) \\ &\times \mathcal{IW}(\mathbf{Q}_n; \upsilon_{n|n-1} + 1, \mathbf{V}_{n|n-1} + (\mathbf{x}_n - \mathbf{A}\mathbf{x}_{n-1})(\cdot)^{\mathrm{T}}),\end{aligned} \tag{4.12}$$

where $\{u_j\}_{j=1}^{J}$ denote the uniformly spaced grid points in the interval $[\max(0, \mu_{\mathcal{G}} - 3\sigma_{\mathcal{G}}),\ \mu_{\mathcal{G}} + 3\sigma_{\mathcal{G}}]$ where $\mu_{\mathcal{G}}$ and $\sigma_{\mathcal{G}}$ are the mean and standard deviation of the density

$\mathcal{G}(\lambda;\nu/2,\nu/2)$ with $\nu = v_{n|n-1} - 2n_x$, respectively, and the weights $\{\alpha_j\}_{j=1}^{J}$ satisfy $\alpha_j \propto \mathcal{G}(u_j;\nu/2,\nu/2)$. The parameter $\overline{\mathbf{Q}}_{n|n-1}$ in (4.12) is defined as

$$\overline{\mathbf{Q}}_{n|n-1} \triangleq \frac{\mathbf{V}_{n|n-1}}{v_{n|n-1} - 2n_x}. \tag{4.13}$$

The second line of (4.12) can be written as

$$\begin{aligned}
&\mathcal{N}(\mathbf{x}_{n-1};\hat{\mathbf{x}}_{n-1|n-1},\mathbf{P}_{n-1|n-1})\mathcal{N}\left(\mathbf{x}_n;\mathbf{A}\mathbf{x}_{n-1},\frac{\overline{\mathbf{Q}}_{n|n-1}}{u_j}\right)\\
&=\mathcal{N}\left(\begin{bmatrix}\mathbf{x}_n\\ \mathbf{x}_{n-1}\end{bmatrix};\begin{bmatrix}\hat{\mathbf{x}}_{n|n-1}\\ \hat{\mathbf{x}}_{n-1|n-1}\end{bmatrix},\begin{bmatrix}\mathbf{P}^j_{n|n-1} & \mathbf{A}\mathbf{P}_{n-1|n-1}\\ \mathbf{P}_{n-1|n-1}\mathbf{A}^{\mathrm{T}} & \mathbf{P}_{n-1|n-1}\end{bmatrix}\right) && (4.14a)\\
&=\mathcal{N}\left(\mathbf{z}_n;\hat{\mathbf{z}},\mathbf{Z}_j\right). && (4.14b)
\end{aligned}$$

where

$$\hat{\mathbf{x}}_{n|n-1} \triangleq \mathbf{A}\hat{\mathbf{x}}_{n-1|n-1}, \tag{4.15a}$$

$$\mathbf{P}^j_{n|n-1} \triangleq \mathbf{A}\mathbf{P}_{n-1|n-1}\mathbf{A}^{\mathrm{T}} + \frac{\overline{\mathbf{Q}}_{n|n-1}}{u_j}, \tag{4.15b}$$

$$\mathbf{z}_n \triangleq \begin{bmatrix}\mathbf{x}_n\\ \mathbf{x}_{n-1}\end{bmatrix}, \tag{4.15c}$$

$$\hat{\mathbf{z}} \triangleq \begin{bmatrix}\hat{\mathbf{x}}_{n|n-1}\\ \hat{\mathbf{x}}_{n-1|n-1}\end{bmatrix}, \tag{4.15d}$$

$$\mathbf{Z}_j \triangleq \begin{bmatrix}\mathbf{P}^j_{n|n-1} & \mathbf{A}\mathbf{P}_{n-1|n-1}\\ \mathbf{P}_{n-1|n-1}\mathbf{A}^{\mathrm{T}} & \mathbf{P}_{n-1|n-1}\end{bmatrix}. \tag{4.15e}$$

Now, we multiply this term with $\mathcal{N}(\mathbf{y}_n;\mathbf{C}\mathbf{x}_n,\mathbf{R}_n)$ and obtain the following.

$$\begin{aligned}
&\mathcal{N}(\mathbf{y}_n;\mathbf{C}\mathbf{x}_n,\mathbf{R}_n)\mathcal{N}\left(\mathbf{z}_n;\hat{\mathbf{z}},\mathbf{Z}_j\right)\\
&=\mathcal{N}\left(\mathbf{y}_n;\overline{\mathbf{C}}\hat{\mathbf{z}},\overline{\mathbf{C}}\mathbf{Z}_j\overline{\mathbf{C}}^{\mathrm{T}}+\mathbf{R}_n\right)\mathcal{N}\left(\mathbf{z}_n;\overline{\mathbf{z}}_j,\overline{\mathbf{Z}}_j\right), && (4.16)
\end{aligned}$$

where

$$\overline{\mathbf{C}} \triangleq \begin{bmatrix}\mathbf{C} & \mathbf{0}\end{bmatrix}, \tag{4.17a}$$

$$\overline{\mathbf{z}}_j = \overline{\mathbf{Z}}_j\left(\mathbf{Z}_j^{-1}\hat{\mathbf{z}} + \overline{\mathbf{C}}^{\mathrm{T}}\mathbf{R}_n^{-1}\mathbf{y}_n\right), \tag{4.17b}$$

$$\overline{\mathbf{Z}}_j = \left(\mathbf{Z}_j^{-1} + \overline{\mathbf{C}}^{\mathrm{T}}\mathbf{R}_n^{-1}\overline{\mathbf{C}}\right)^{-1}. \tag{4.17c}$$

The right hand side of (4.12) can be written by using (4.16) as follows:

$$p(\mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{Q}_n | \mathbf{y}_{0:n}) \propto$$
$$\sum_{j=1}^{J} \alpha_j \mathcal{N}\left(\mathbf{y}_n; \overline{\mathbf{C}}\hat{\mathbf{z}}, \overline{\mathbf{C}}\mathbf{Z}_j\overline{\mathbf{C}}^{\mathrm{T}} + \mathbf{R}_n\right) \mathcal{N}\left(\mathbf{z}_n; \overline{\mathbf{z}}_j, \overline{\mathbf{Z}}_j\right)$$
$$\times \mathcal{IW}(\mathbf{Q}_n; \upsilon_{n|n-1} + 1, \mathbf{V}_{n|n-1} + (\mathbf{x}_n - \mathbf{A}\mathbf{x}_{n-1})(\cdot)^{\mathrm{T}}). \tag{4.18}$$

To simplify the above expression, $\overline{\alpha}_j$ is defined as follows:

$$\overline{\alpha}_j \triangleq \frac{\alpha_j \mathcal{N}\left(\mathbf{y}_n; \overline{\mathbf{C}}\hat{\mathbf{z}}, \overline{\mathbf{C}}\mathbf{Z}_j\overline{\mathbf{C}}^{\mathrm{T}} + \mathbf{R}_n\right)}{\sum_{j=1}^{J} \alpha_j \mathcal{N}\left(\mathbf{y}_n; \overline{\mathbf{C}}\hat{\mathbf{z}}, \overline{\mathbf{C}}\mathbf{Z}_j\overline{\mathbf{C}}^{\mathrm{T}} + \mathbf{R}_n\right)}. \tag{4.19}$$

When we insert $\overline{\alpha}_j$ into (4.18), we obtain

$$p(\mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{Q}_n | \mathbf{y}_{0:n}) \propto \sum_{j=1}^{J} \overline{\alpha}_j \mathcal{N}\left(\mathbf{z}_n; \overline{\mathbf{z}}_j, \overline{\mathbf{Z}}_j\right)$$
$$\times \mathcal{IW}(\mathbf{Q}_n; \upsilon_{n|n-1} + 1, \mathbf{V}_{n|n-1} + (\mathbf{x}_n - \mathbf{A}\mathbf{x}_{n-1})(\cdot)^{\mathrm{T}}). \tag{4.20}$$

By defining the vector $\mathbf{w}_n \triangleq \mathbf{x}_n - \mathbf{A}\mathbf{x}_{n-1}$, we can write $p(\mathbf{w}_n, \mathbf{Q}_n | \mathbf{y}_{0:n})$ based on (4.20) as

$$p(\mathbf{w}_n, \mathbf{Q}_n | \mathbf{y}_{0:n}) = \sum_{j=1}^{J} \overline{\alpha}_j \mathcal{N}(\mathbf{w}_n; \overline{\mathbf{w}}_j, \overline{\mathbf{W}}_j)$$
$$\times \mathcal{IW}(\mathbf{Q}_n; \upsilon_{n|n-1} + 1, \mathbf{V}_{n|n-1} + \mathbf{w}_n\mathbf{w}_n^{\mathrm{T}}), \tag{4.21}$$

where

$$\overline{\mathbf{w}}_j = \begin{bmatrix} \mathcal{I} & -\mathbf{A} \end{bmatrix} \overline{\mathbf{z}}_j, \tag{4.22a}$$
$$\overline{\mathbf{W}}_j = \begin{bmatrix} \mathcal{I} & -\mathbf{A} \end{bmatrix} \overline{\mathbf{Z}}_j \begin{bmatrix} \mathcal{I} \\ -\mathbf{A}^{\mathrm{T}} \end{bmatrix}. \tag{4.22b}$$

The marginal posterior $p(\mathbf{Q}_n | \mathbf{y}_{0:n})$ can now be obtained by marginalizing $p(\mathbf{w}_n, \mathbf{Q}_n | \mathbf{y}_{0:n})$ with respect to $\mathbf{w}_n$ as

$$p(\mathbf{Q}_n | \mathbf{y}_{0:n}) = \int \sum_{j=1}^{J} \overline{\alpha}_j \mathcal{N}(\mathbf{w}_n; \overline{\mathbf{w}}_j, \overline{\mathbf{W}}_j)$$
$$\times \mathcal{IW}(\mathbf{Q}_n; \upsilon_{n|n-1} + 1, \mathbf{V}_{n|n-1} + \mathbf{w}_n\mathbf{w}_n^{\mathrm{T}})\, \mathrm{d}\mathbf{w}_n \tag{4.23}$$
$$= \int r(\mathbf{w}_n) \mathcal{IW}(\mathbf{Q}_n; \upsilon_{n|n-1} + 1, \mathbf{V}(\mathbf{w}_n))\, \mathrm{d}\mathbf{w}_n, \tag{4.24}$$

where

$$r(\mathbf{w}_n) \triangleq \sum_{j=1}^{J} \overline{\alpha}_j \mathcal{N}(\mathbf{w}_n; \overline{\mathbf{w}}_j, \overline{\mathbf{W}}_j), \tag{4.25a}$$

$$\mathbf{V}(\mathbf{w}_n) \triangleq \mathbf{V}_{n|n-1} + \mathbf{w}_n \mathbf{w}_n^{\mathrm{T}}. \tag{4.25b}$$

Using the moment matching based approximation given in Section 4.10.2 on (4.24), the marginal distribution $p(\mathbf{Q}_n|\mathbf{y}_{0:n})$ is given as

$$p(\mathbf{Q}_n|\mathbf{y}_{0:n}) \approx \mathcal{IW}(\mathbf{Q}_n; \upsilon_{n|n}, \mathbf{V}_{n|n}), \tag{4.26}$$

where

$$\upsilon_{n|n} = 2n_x + 2 + \frac{2(\upsilon_{n|n-1} - 2n_x - 1)}{2\alpha + (1-\alpha)(\upsilon_{n|n-1} - 2n_x - 1)}, \tag{4.27a}$$

$$\mathbf{V}_{n|n} = \frac{\upsilon_{n|n} - 2n_x - 2}{\upsilon_{n|n-1} - 2n_x - 1} \mathrm{E}_r \left[\mathbf{V}(\mathbf{w}_n)\right], \tag{4.27b}$$

$$\alpha = \frac{\sum_{m=1}^{n_x} \mathrm{E}_r^2 \left[\mathbf{V}_{mm}(\mathbf{w}_n)\right]}{\sum_{m=1}^{n_x} \mathrm{E}_r \left[\mathbf{V}_{mm}^2(\mathbf{w}_n)\right]}. \tag{4.27c}$$

The expectations required in (4.27) are given in the following equations. The expectation of $\mathbf{V}(\mathbf{w}_n)$ is found as

$$\mathrm{E}_r[\mathbf{V}(\mathbf{w}_n)] = \mathrm{E}_r\left[\mathbf{V}_{n|n-1} + \mathbf{w}_n \mathbf{w}_n^{\mathrm{T}}\right] \tag{4.28a}$$

$$= \mathbf{V}_{n|n-1} + \mathrm{E}_r\left[\mathbf{w}_n \mathbf{w}_n^{\mathrm{T}}\right] \tag{4.28b}$$

$$= \mathbf{V}_{n|n-1} + \sum_{j=1}^{J} \overline{\alpha}_j \left[\overline{\mathbf{W}}_j + \overline{\mathbf{w}}_j \overline{\mathbf{w}}_j^{\mathrm{T}}\right]. \tag{4.28c}$$

The $m^{th}$ diagonal element of $\mathrm{E}_r[\mathbf{V}(\mathbf{w}_n)]$, i.e., $\mathrm{E}_r[\mathbf{V}_{mm}(\mathbf{w}_n)]$, is given as

$$\mathrm{E}_r[\mathbf{V}_{mm}(\mathbf{w}_n)] = (\mathbf{V}_{n|n-1})_{mm} + \sum_{j=1}^{J} \overline{\alpha}_j \left[(\overline{\mathbf{W}}_j)_{mm} + (\overline{\mathbf{w}}_j)_m^2\right]. \tag{4.29}$$

The last expectation $\mathrm{E}_r[\mathbf{V}_{mm}^2(\mathbf{w}_n)]$ is calculated as

$$\mathrm{E}_r[\mathbf{V}_{mm}^2(\mathbf{w}_n)] = \mathrm{E}_r\left[(\mathbf{V}_{n|n-1})_{mm}^2 + 2(\mathbf{V}_{n|n-1})_{mm}(\mathbf{w}_n)_m^2 + (\mathbf{w}_n)_m^4\right] \tag{4.30a}$$

$$=(\mathbf{V}_{n|n-1})_{mm}^2 + 2(\mathbf{V}_{n|n-1})_{mm} \mathrm{E}_r\left[(\mathbf{w}_n)_m^2\right] + \mathrm{E}_r\left[(\mathbf{w}_n)_m^4\right] \tag{4.30b}$$

$$=(\mathbf{V}_{n|n-1})_{mm}^2 + 2(\mathbf{V}_{n|n-1})_{mm} \sum_{j=1}^{J} \overline{\alpha}_j \left((\overline{\mathbf{w}}_j)_m^2 + (\overline{\mathbf{W}}_j)_{mm}\right)$$
$$+ \sum_{j=1}^{J} \overline{\alpha}_j \left((\overline{\mathbf{w}}_j)_m^4 + 6(\overline{\mathbf{w}}_j)_m^2 (\overline{\mathbf{W}}_j)_{mm} + 3(\overline{\mathbf{W}}_j)_{mm}^2\right). \tag{4.30c}$$

The filtered distribution of the state $\mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n})$ is found by integrating (4.12) first with respect to $\mathbf{Q}_n$ then with respect to $\mathbf{x}_{n-1}$ as follows:

$$p(\mathbf{x}_n|\mathbf{y}_{0:n}) \propto \int \sum_{j=1}^{J} \alpha_j \mathcal{N}(\mathbf{y}_n; \mathbf{C}\mathbf{x}_n, \mathbf{R}_n)$$

$$\times \mathcal{N}(\mathbf{x}_{n-1}, \hat{\mathbf{x}}_{n-1|n-1}, \mathbf{P}_{n-1|n-1}) \mathcal{N}\left(\mathbf{x}_n; \mathbf{A}\mathbf{x}_{n-1}, \frac{\overline{\mathbf{Q}}_{n|n-1}}{u_j}\right) \mathrm{d}\mathbf{x}_{n-1} \tag{4.31a}$$

$$= \sum_{j=1}^{J} \alpha_j \mathcal{N}(\mathbf{y}_n; \mathbf{C}\mathbf{x}_n, \mathbf{R}_n) \mathcal{N}\left(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n-1}, \mathbf{P}^j_{n|n-1}\right) \tag{4.31b}$$

$$= \sum_{j=1}^{J} \pi_j \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}^j_{n|n}, \hat{\mathbf{P}}^j_{n|n}), \tag{4.31c}$$

where the parameters $\pi_j$, $\hat{\mathbf{x}}^j_{n|n}$ and $\hat{\mathbf{P}}^j_{n|n}$ are given as

$$\pi_j \propto \alpha_j \mathcal{N}\left(\mathbf{y}_n; \mathbf{C}\hat{\mathbf{x}}_{n|n-1}, \mathbf{C}\mathbf{P}^j_{n|n-1}\mathbf{C}^{\mathrm{T}} + \mathbf{R}_n\right), \tag{4.32a}$$

$$\hat{\mathbf{x}}^j_{n|n} = \hat{\mathbf{P}}^j_{n|n} \left(\left(\mathbf{P}^j_{n|n-1}\right)^{-1} \hat{\mathbf{x}}_{n-1|n-1} + \mathbf{C}^{\mathrm{T}}\mathbf{R}_n^{-1}\mathbf{y}_n\right), \tag{4.32b}$$

$$\hat{\mathbf{P}}^j_{n|n} = \left(\left(\mathbf{P}^j_{n|n-1}\right)^{-1} + \mathbf{C}^{\mathrm{T}}\mathbf{R}_n^{-1}\mathbf{C}\right)^{-1}. \tag{4.32c}$$

Finally we can get $p(\mathbf{x}_n|\mathbf{y}_{0:n}) \approx \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n})$ where the mean $\hat{\mathbf{x}}_{n|n}$ and the covariance $\mathbf{P}_{n|n}$ are found by moment matching as

$$\hat{\mathbf{x}}_{n|n} = \sum_{j=1}^{J} \pi_j \hat{\mathbf{x}}^j_{n|n}, \tag{4.33a}$$

$$\mathbf{P}_{n|n} = \sum_{j=1}^{J} \pi_j \left(\hat{\mathbf{P}}^j_{n|n} + (\hat{\mathbf{x}}^j_{n|n} - \hat{\mathbf{x}}_{n|n})(\hat{\mathbf{x}}^j_{n|n} - \hat{\mathbf{x}}_{n|n})^{\mathrm{T}}\right). \tag{4.33b}$$

The pseudo-code of a single step of the resulting filter is provided in Algorithm 2.

## 4.4 Bayesian Filtering with Unknown MNC

In this section, we investigate Bayesian filtering with unknown MNC when PNCs are known. Similar to the solution of unknown PNC case, we first write the joint posterior distribution of state and measurement noise covariance. We approximate the filtered posterior $p(\mathbf{x}_n, \mathbf{R}_n|\mathbf{y}_{0:n})$ as

$$p(\mathbf{x}_n, \mathbf{R}_n|\mathbf{y}_{0:n}) \approx \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n}) \mathcal{IW}(\mathbf{R}_n; w_{n|n}, \mathbf{W}_{n|n}). \tag{4.34}$$

**Algorithm 2** Filtering with Unknown PNC

**Inputs:** $\mathbf{A}, \mathbf{C}, \mathbf{R}_n, \mathbf{y}_n, J, \gamma_Q, \hat{\mathbf{x}}_{n-1|n-1}, \mathbf{P}_{n-1|n-1}, v_{n-1|n-1}, \mathbf{V}_{n-1|n-1}$

Calculate $v_{n|n-1}$ and $\mathbf{V}_{n|n-1}$ using (4.7).

Calculate the grid-points $\{u_j\}_{j=1}^{J}$ and their weights $\{\alpha_j\}_{j=1}^{J}$ from Gamma distribution with parameters $\alpha = \beta = (v_{n|n-1} - 2n_x)/2$.

Calculate $\hat{\mathbf{x}}_{n|n-1}$ and $\{\hat{\mathbf{P}}_{n|n-1}^{j}\}_{j=1}^{J}$ using (4.15).

Use (4.32) to calculate $\{\hat{\mathbf{x}}_{n|n}^{j}, \hat{\mathbf{P}}_{n|n}^{j}\}_{j=1}^{J}$.

Merge $\{\hat{\mathbf{x}}_{n|n}^{j}, \hat{\mathbf{P}}_{n|n}^{j}\}_{j=1}^{J}$ using (4.33) to obtain $\hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n}$.

Calculate moments $\mathrm{E}_r[\mathbf{V}(\mathbf{w}_n)], \mathrm{E}_r[\mathbf{V}_{mm}(\mathbf{w}_n)]$ and $\mathrm{E}_r[\mathbf{V}_{mm}^2(\mathbf{w}_n)]$ using equations (4.28), (4.29), and (4.30), respectively

Calculate $v_{n|n}$ and $\mathbf{V}_{n|n}$ using (4.27).

**Outputs:** $\hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n}, v_{n|n}, \mathbf{V}_{n|n}$

To obtain the parameters of the joint posterior distribution, we first write $p(\mathbf{x}_n, \mathbf{R}_n|\mathbf{y}_{0:n-1})$ by using Chapman-Kolmogorov equation as follows:

$$
\begin{aligned}
p(\mathbf{x}_n, \mathbf{R}_n|\mathbf{y}_{0:n-1}) = &\int\int p(\mathbf{x}_n, \mathbf{R}_n|\mathbf{x}_{n-1}, \mathbf{R}_{n-1}) \\
&\times p(\mathbf{x}_{n-1}, \mathbf{R}_{n-1}|\mathbf{y}_{0:n-1})\, \mathrm{d}\mathbf{x}_{n-1}\, \mathrm{d}\mathbf{R}_{n-1} \qquad (4.35a)\\
= &\int\int \mathcal{N}(\mathbf{x}_n; \mathbf{A}\mathbf{x}_{n-1}, \mathbf{Q}_n)\mathcal{N}(\mathbf{x}_{n-1}; \hat{\mathbf{x}}_{n-1|n-1}, \mathbf{P}_{n-1|n-1}) \\
&\times p(\mathbf{R}_n|\mathbf{R}_{n-1})\mathcal{IW}(\mathbf{R}_{n-1}; w_{n-1|n-1}, \mathbf{W}_{n-1|n-1})\, \mathrm{d}\mathbf{x}_{n-1}\, \mathrm{d}\mathbf{R}_{n-1} \qquad (4.35b)\\
= &\mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n-1}, \mathbf{P}_{n|n-1})\mathcal{IW}(\mathbf{R}_n; w_{n|n-1}, \mathbf{W}_{n|n-1}), \qquad (4.35c)
\end{aligned}
$$

where

$$
\begin{aligned}
\hat{\mathbf{x}}_{n|n-1} &\triangleq \mathbf{A}\hat{\mathbf{x}}_{n-1|n-1}, && (4.36a)\\
\mathbf{P}_{n|n-1} &\triangleq \mathbf{A}\mathbf{P}_{n-1|n-1}\mathbf{A}^{\mathrm{T}} + \mathbf{Q}_n, && (4.36b)\\
w_{n|n-1} &\triangleq \gamma_R w_{n-1|n-1} + (1-\gamma_R)(2n_y + 2), && (4.36c)\\
\mathbf{W}_{n|n-1} &\triangleq \gamma_R \mathbf{W}_{n-1|n-1}. && (4.36d)
\end{aligned}
$$

The updated posterior distribution $p(\mathbf{x}_n, \mathbf{R}_n|\mathbf{y}_{0:n})$ is found by using Bayes' rule as follows:

$$
\begin{aligned}
p(\mathbf{x}_n, \mathbf{R}_n|\mathbf{y}_{0:n}) &\propto p(\mathbf{y}_n|\mathbf{x}_n, \mathbf{R}_n)p(\mathbf{x}_n, \mathbf{R}_n|\mathbf{y}_{0:n-1}) && (4.37a)\\
&= \mathcal{N}(\mathbf{y}_n; \mathbf{C}\mathbf{x}_n, \mathbf{R}_n)\mathcal{IW}(\mathbf{R}_n; w_{n|n-1}, \mathbf{W}_{n|n-1})\mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n-1}, \mathbf{P}_{n|n-1}). && (4.37b)
\end{aligned}
$$

The posterior distributions of $\mathbf{x}_n$ and $\mathbf{R}_n$ cannot be found separately since the $\mathbf{R}_n$ term appears in both Gaussian and IW distributions. To solve this problem, we write the first two terms given in (4.37b) as the multiplication of Student's *t* and IW distribution.

$$\begin{aligned}\mathcal{N}(\mathbf{y}_n;&\mathbf{C}\mathbf{x}_n,\mathbf{R}_n)\mathcal{IW}(\mathbf{R}_n;w_{n|n-1},\mathbf{W}_{n|n-1})\\ &\propto \text{St}\left(\mathbf{y}_n;\mathbf{C}\mathbf{x}_n,\frac{\mathbf{W}_{n|n-1}}{w_{n|n-1}-2n_y},w_{n|n-1}-2n_y\right)\\ &\quad\times\mathcal{IW}(\mathbf{R}_n;w_{n|n-1}+1,\mathbf{W}_{n|n-1}+(\mathbf{y}_n-\mathbf{C}\mathbf{x}_n)(\mathbf{y}_n-\mathbf{C}\mathbf{x}_n)^{\mathrm{T}}).\end{aligned} \tag{4.38}$$

By using (4.11), the *t*-distribution can be approximated as

$$\text{St}\left(\mathbf{y}_n;\mathbf{C}\mathbf{x}_n,\frac{\mathbf{W}_{n|n-1}}{w_{n|n-1}-2n_y},w_{n|n-1}-2n_y\right)\approx\sum_{j=1}^{J}\alpha_j\mathcal{N}\left(\mathbf{y}_n;\mathbf{C}\mathbf{x}_n,\frac{\overline{\mathbf{R}}_{n|n-1}}{u_j}\right), \tag{4.39}$$

where $\{u_j\}_{j=1}^{J}$ denote the uniformly spaced grid points in the interval $(\max(0,\mu_{\mathcal{G}}-3\sigma_{\mathcal{G}}),\ \mu_{\mathcal{G}}+3\sigma_{\mathcal{G}}]$ where $\mu_{\mathcal{G}}$ and $\sigma_{\mathcal{G}}$ are the mean and standard deviation of the density $\mathcal{G}(\lambda;\nu/2,\nu/2)$ with $\nu=w_{n|n-1}-2n_y$, respectively, and the weights $\{\alpha_j\}_{j=1}^{J}$ satisfy $\alpha_j\propto\mathcal{G}(u_j;\nu/2,\nu/2)$. The parameter $\overline{\mathbf{R}}_{n|n-1}$ in (4.39) is defined as

$$\overline{\mathbf{R}}_{n|n-1}\triangleq\frac{\mathbf{W}_{n|n-1}}{w_{n|n-1}-2n_y}. \tag{4.40}$$

We can now write the updated posterior as

$$\begin{aligned}p(\mathbf{x}_n,\mathbf{R}_n|\mathbf{y}_{0:n})&\propto\sum_{j=1}^{J}\alpha_j\mathcal{N}\left(\mathbf{y}_n;\mathbf{C}\mathbf{x}_n,\frac{\overline{\mathbf{R}}_{n|n-1}}{u_j}\right)\\ &\times\mathcal{IW}(\mathbf{R}_n;w_{n|n-1}+1,\mathbf{W}_{n|n-1}+(\mathbf{y}_n-\mathbf{C}\mathbf{x}_n)(\cdot)^{\mathrm{T}})\mathcal{N}(\mathbf{x}_n;\hat{\mathbf{x}}_{n|n-1},\mathbf{P}_{n|n-1})\end{aligned} \tag{4.41a}$$

$$=\sum_{j=1}^{J}\overline{\alpha}_j\mathcal{N}(\mathbf{x}_n;\hat{\mathbf{x}}^j_{n|n},\hat{\mathbf{P}}^j_{n|n})\mathcal{IW}(\mathbf{R}_n;w_{n|n-1}+1,\mathbf{W}_{n|n-1}+(\mathbf{y}_n-\mathbf{C}\mathbf{x}_n)(\cdot)^{\mathrm{T}}), \tag{4.41b}$$

where

$$\overline{\alpha}_j\propto\alpha_j\mathcal{N}\left(\mathbf{y}_n;\mathbf{C}\hat{\mathbf{x}}_{n|n-1},\mathbf{C}\mathbf{P}_{n|n-1}\mathbf{C}^{\mathrm{T}}+\frac{\overline{\mathbf{R}}_{n|n-1}}{u_j}\right), \tag{4.42a}$$

$$\hat{\mathbf{x}}^j_{n|n}=\hat{\mathbf{P}}^j_{n|n}\left(\mathbf{P}^{-1}_{n|n-1}\hat{\mathbf{x}}_{n|n-1}+u_j\mathbf{C}^{\mathrm{T}}\overline{\mathbf{R}}^{-1}_{n|n-1}\mathbf{y}_n\right), \tag{4.42b}$$

$$\hat{\mathbf{P}}^j_{n|n}=\left(\mathbf{P}^{-1}_{n|n-1}+u_j\mathbf{C}^{\mathrm{T}}\overline{\mathbf{R}}^{-1}_{n|n-1}\mathbf{C}\right)^{-1}. \tag{4.42c}$$

Integrating $p(\mathbf{x}_n, \mathbf{R}_n|\mathbf{y}_{0:n})$ with respect to $\mathbf{R}_n$, the marginal posterior $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ can be calculated as

$$p(\mathbf{x}_n|\mathbf{y}_{0:n}) = \sum_{j=1}^{J} \overline{\alpha}_j \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}^j_{n|n}, \hat{\mathbf{P}}^j_{n|n}) \tag{4.43a}$$

$$\approx \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n}), \tag{4.43b}$$

where

$$\hat{\mathbf{x}}_{n|n} = \sum_{j=1}^{J} \overline{\alpha}_j \hat{\mathbf{x}}^j_{n|n}, \tag{4.44a}$$

$$\mathbf{P}_{n|n} = \sum_{j=1}^{J} \overline{\alpha}_j \left( \hat{\mathbf{P}}^j_{n|n} + (\hat{\mathbf{x}}^j_{n|n} - \hat{\mathbf{x}}_{n|n})(\hat{\mathbf{x}}^j_{n|n} - \hat{\mathbf{x}}_{n|n})^{\mathrm{T}} \right). \tag{4.44b}$$

The marginal posterior $p(\mathbf{R}_n|\mathbf{y}_{0:n})$ can be obtained by integrating $p(\mathbf{x}_n, \mathbf{R}_n|\mathbf{y}_{0:n})$ with respect to $\mathbf{x}_n$ as

$$p(\mathbf{R}_n|\mathbf{y}_{0:n}) = \int \sum_{j=1}^{J} \overline{\alpha}_j \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}^j_{n|n}, \hat{\mathbf{P}}^j_{n|n})$$

$$\times \mathcal{IW}(\mathbf{R}_n; w_{n|n-1} + 1, \mathbf{W}_{n|n-1} + (\mathbf{y}_n - \mathbf{C}\mathbf{x}_n)(\cdot)^{\mathrm{T}})\ \mathrm{d}\mathbf{x}_n \tag{4.45a}$$

$$= \int r(\mathbf{x}_n) \mathcal{IW}(\mathbf{R}_n; w_{n|n-1} + 1, \mathbf{W}(\mathbf{x}_n))\ \mathrm{d}\mathbf{x}_n, \tag{4.45b}$$

where

$$r(\mathbf{x}_n) \triangleq \sum_{j=1}^{J} \overline{\alpha}_j \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}^j_{n|n}, \hat{\mathbf{P}}^j_{n|n}), \tag{4.46a}$$

$$\mathbf{W}(\mathbf{x}_n) \triangleq \mathbf{W}_{n|n-1} + (\mathbf{y}_n - \mathbf{C}\mathbf{x}_n)(\mathbf{y}_n - \mathbf{C}\mathbf{x}_n)^{\mathrm{T}}. \tag{4.46b}$$

Using the moment matching based approximation used in Section 4.10.2, the marginal distribution $p(\mathbf{R}_n|\mathbf{y}_{0:n})$ is given as

$$p(\mathbf{R}_n|\mathbf{y}_{0:n}) \approx \mathcal{IW}(\mathbf{R}_n; w_{n|n}, \mathbf{W}_{n|n}), \tag{4.47}$$

where

$$w_{n|n} = 2n_y + 2 + \frac{2(w_{n|n-1} - 2n_y - 1)}{2\alpha + (1-\alpha)(w_{n|n-1} - 2n_y - 1)}, \tag{4.48a}$$

$$\mathbf{W}_{n|n} = \frac{w_{n|n} - 2n_y - 2}{w_{n|n-1} - 2n_y - 1} \mathrm{E}_r\left[\mathbf{W}(\mathbf{x}_n)\right], \tag{4.48b}$$

$$\alpha = \frac{\sum_{m=1}^{n_y} \mathrm{E}_r^2\left[\mathbf{W}_{mm}(\mathbf{x}_n)\right]}{\sum_{m=1}^{n_y} \mathrm{E}_r\left[\mathbf{W}^2_{mm}(\mathbf{x}_n)\right]}. \tag{4.48c}$$

The expectations required in (4.48) are given as

$$\mathrm{E}_r[\mathbf{W}(\mathbf{x}_n)] = \mathbf{W}_{n|n-1} + \sum_{j=1}^{J} \overline{\alpha}_j \left( (\mathbf{y}_n - \mathbf{C}\hat{\mathbf{x}}^j_{n|n})(\cdot)^{\mathrm{T}} + \mathbf{C}\hat{\mathbf{P}}^j_{n|n}\mathbf{C}^{\mathrm{T}} \right), \tag{4.49a}$$

$$\mathrm{E}_r[\mathbf{W}_{mm}(\mathbf{x}_n)] = (\mathbf{W}_{n|n-1})_{mm} + \sum_{j=1}^{J} \overline{\alpha}_j \left( (\mathbf{y}_n - \mathbf{C}\hat{\mathbf{x}}^j_{n|n})^2_m + (\mathbf{C}\hat{\mathbf{P}}^j_{n|n}\mathbf{C}^{\mathrm{T}})_{mm} \right), \tag{4.49b}$$

$$\begin{aligned}\mathrm{E}_r[\mathbf{W}^2_{mm}(\mathbf{x}_n)] &= (\mathbf{W}_{n|n-1})^2_{mm} + 2(\mathbf{W}_{n|n-1})_{mm} \\ &\times \sum_{j=1}^{J} \overline{\alpha}_j \left( (\mathbf{y}_n - \mathbf{C}\hat{\mathbf{x}}^j_{n|n})^2_m + (\mathbf{C}\hat{\mathbf{P}}^j_{n|n}\mathbf{C}^{\mathrm{T}})_{mm} \right) \\ &+ \sum_{j=1}^{J} \overline{\alpha}_j \Big( (\mathbf{y}_n - \mathbf{C}\hat{\mathbf{x}}^j_{n|n})^4_m + 6(\mathbf{y}_n - \mathbf{C}\hat{\mathbf{x}}^j_{n|n})^2_m \\ &\times (\mathbf{C}\hat{\mathbf{P}}^j_{n|n}\mathbf{C}^{\mathrm{T}})_{mm} + 3(\mathbf{C}\hat{\mathbf{P}}^j_{n|n}\mathbf{C}^{\mathrm{T}})^2_{mm} \Big).\end{aligned} \tag{4.49c}$$

The pseudo-code of a single step of the resulting filter is provided in Algorithm 3.

**Algorithm 3** Filtering with Unknown Measurement Noise Covariance

1: **Inputs:** $\mathbf{A}, \mathbf{C}, \mathbf{Q}_n, \mathbf{y}_n, J, \gamma_R, \hat{\mathbf{x}}_{n-1|n-1}, \mathbf{P}_{n-1|n-1}, w_{n-1|n-1}, \mathbf{W}_{n-1|n-1}$
2: Calculate $\hat{\mathbf{x}}_{n|n-1}, \hat{\mathbf{P}}_{n|n-1}, w_{n|n-1}, \mathbf{W}_{n|n-1}$ using (4.36).
3: Calculate the grid-points $\{u_j\}_{j=1}^J$ and their weights $\{\alpha_j\}_{j=1}^J$ from Gamma distribution with parameters $\alpha = \beta = (w_{n|n-1} - 2n_y)/2$.
4: Calculate $\{\hat{\mathbf{x}}^j_{n|n}, \hat{\mathbf{P}}^j_{n|n}\}_{j=1}^J$ using (4.42).
5: Merge $\{\hat{\mathbf{x}}^j_{n|n}, \hat{\mathbf{P}}^j_{n|n}\}_{j=1}^J$ using (4.44) to obtain $\hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n}$.
6: Calculate moments $\mathrm{E}_r[\mathbf{W}(\mathbf{x}_n)], \mathrm{E}_r[\mathbf{W}_{mm}(\mathbf{x}_n)]$ and $\mathrm{E}_r[\mathbf{W}^2_{mm}(\mathbf{x}_n)]$ using equations (4.49).
7: Calculate $w_{n|n}$ and $\mathbf{W}_{n|n}$ using (4.48).
8: **Outputs:** $\hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n}, w_{n|n}, \mathbf{W}_{n|n}$

### 4.5 Bayesian Smoothing with Unknown PNC

In this section, smoothing with unknown PNC problem is investigated when MNCs are known. The smoothed distribution $p(\mathbf{x}_n, \mathbf{Q}_n|\mathbf{y}_{0:N})$ can be written as follows:

$$p(\mathbf{x}_n, \mathbf{Q}_n|\mathbf{y}_{0:N}) = \int\int \frac{p(\mathbf{x}_{n+1}, \mathbf{Q}_{n+1}|\mathbf{x}_n, \mathbf{Q}_n)p(\mathbf{x}_n, \mathbf{Q}_n|\mathbf{y}_{0:n})}{p(\mathbf{x}_{n+1}, \mathbf{Q}_{n+1}|\mathbf{y}_{0:n})}$$

$$\times\, p(\mathbf{x}_{n+1}, \mathbf{Q}_{n+1}|\mathbf{y}_{0:N})\ \mathrm{d}\mathbf{x}_{n+1}\ \mathrm{d}\mathbf{Q}_{n+1}. \tag{4.50}$$

We can write the terms of the integrand above as follows:

$$p(\mathbf{x}_{n+1}, \mathbf{Q}_{n+1}|\mathbf{x}_n, \mathbf{Q}_n) = \mathcal{N}(\mathbf{x}_{n+1}; \mathbf{A}\mathbf{x}_n, \mathbf{Q}_{n+1})p(\mathbf{Q}_{n+1}|\mathbf{Q}_n), \tag{4.51a}$$

$$p(\mathbf{x}_n, \mathbf{Q}_n|\mathbf{y}_{0:n}) = \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n})\mathcal{IW}(\mathbf{Q}_n; \upsilon_{n|n}, \mathbf{V}_{n|n}), \tag{4.51b}$$

$$p(\mathbf{x}_{n+1}, \mathbf{Q}_{n+1}|\mathbf{y}_{0:n}) = \mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|n}, \mathbf{P}_{n+1|n})\mathcal{IW}(\mathbf{Q}_{n+1}; \upsilon_{n+1|n}, \mathbf{V}_{n+1|n}), \tag{4.51c}$$

$$p(\mathbf{x}_{n+1}, \mathbf{Q}_{n+1}|\mathbf{y}_{0:N}) = \mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|N}, \mathbf{P}_{n+1|N})\mathcal{IW}(\mathbf{Q}_{n+1}; \upsilon_{n+1|N}, \mathbf{V}_{n+1|N}), \tag{4.51d}$$

where

$$\hat{\mathbf{x}}_{n+1|n} = \mathbf{A}\hat{\mathbf{x}}_{n|n}, \tag{4.52a}$$

$$\mathbf{P}_{n+1|n} = \mathbf{A}\mathbf{P}_{n|n}\mathbf{A}^{\mathrm{T}} + \mathbf{Q}_{n+1}. \tag{4.52b}$$

Substituting the terms in (4.51) into (4.50) would give

$$\begin{aligned} p(\mathbf{x}_n, \mathbf{Q}_n|\mathbf{y}_{0:N}) = &\int\int \mathcal{N}(\mathbf{x}_{n+1}; \mathbf{A}\mathbf{x}_n, \mathbf{Q}_{n+1})p(\mathbf{Q}_{n+1}|\mathbf{Q}_n) \\ &\times \frac{\mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n})\mathcal{IW}(\mathbf{Q}_n; \upsilon_{n|n}, \mathbf{V}_{n|n})\mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|N}, \mathbf{P}_{n+1|N})}{\mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|n}, \mathbf{P}_{n+1|n})\mathcal{IW}(\mathbf{Q}_{n+1}; \upsilon_{n+1|n}, \mathbf{V}_{n+1|n})} \\ &\times \mathcal{IW}(\mathbf{Q}_{n+1}; \upsilon_{n+1|N}, \mathbf{V}_{n+1|N})\ \mathrm{d}\mathbf{x}_{n+1}\ \mathrm{d}\mathbf{Q}_{n+1}. \end{aligned} \tag{4.53}$$

Unfortunately, the integral cannot be evaluated analytically due to the form of the exact predicted density $p(\mathbf{x}_{n+1}, \mathbf{Q}_{n+1}|\mathbf{y}_{0:n})$. Hence, the following factorized approximation is found for $p(\mathbf{x}_{n+1}, \mathbf{Q}_{n+1}|\mathbf{y}_{0:n})$ instead of the exact density.

$$p(\mathbf{x}_{n+1}, \mathbf{Q}_{n+1}|\mathbf{y}_{0:n}) \approx p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n})p(\mathbf{Q}_{n+1}|\mathbf{y}_{0:n}) \tag{4.54a}$$

$$\begin{aligned} &= \mathcal{N}(\mathbf{x}_{n+1}, \hat{\mathbf{x}}_{n+1|n}, \mathbf{A}\mathbf{P}_{n|n}\mathbf{A}^{\mathrm{T}} + \hat{\mathbf{Q}}_{n+1|n}) \\ &\quad \times \mathcal{IW}(\mathbf{Q}_{n+1}; \upsilon_{n+1|n}, \mathbf{V}_{n+1|n}), \end{aligned} \tag{4.54b}$$

where $\hat{\mathbf{Q}}_{n+1|n} = \mathbf{V}_{n+1|n}/(\upsilon_{n+1|n} - 2n_x - 2)$, a derivation of which is given in Section 4.10.3. Substituting the approximation (4.54b) into the smoothed distribution (4.53) we get

$$p(\mathbf{x}_n, \mathbf{Q}_n|\mathbf{y}_{0:N}) = \int\int \frac{\mathcal{N}(\mathbf{x}_{n+1}; \mathbf{A}\mathbf{x}_n, \mathbf{Q}_{n+1})p(\mathbf{Q}_{n+1}|\mathbf{Q}_n)}{\mathcal{IW}(\mathbf{Q}_{n+1}; \upsilon_{n+1|n}, \mathbf{V}_{n+1|n})}$$

$$\times \frac{\mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n})\mathcal{IW}(\mathbf{Q}_n; \upsilon_{n|n}, \mathbf{V}_{n|n})}{\mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|n}, \mathbf{A}\mathbf{P}_{n|n}\mathbf{A}^{\mathrm{T}} + \hat{\mathbf{Q}}_{n+1|n})}\mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|N}, \mathbf{P}_{n+1|N})$$
$$\times \mathcal{IW}(\mathbf{Q}_{n+1}; \upsilon_{n+1|N}, \mathbf{V}_{n+1|N})\, \mathrm{d}\mathbf{x}_{n+1}\, \mathrm{d}\mathbf{Q}_{n+1}. \tag{4.55}$$

To make the Gaussian distribution $\mathcal{N}(\mathbf{x}_{n+1}; \mathbf{A}\mathbf{x}_n, \mathbf{Q}_{n+1})$ independent of the term $\mathbf{Q}_{n+1}$, the multiplication of the terms $\mathcal{N}(\mathbf{x}_{n+1}; \mathbf{A}\mathbf{x}_n, \mathbf{Q}_{n+1})$ and $\mathcal{IW}(\mathbf{Q}_{n+1}; \upsilon_{n+1|N}, \mathbf{V}_{n+1|N})$ are rewritten as the multiplication of Student's *t* and inverse Wishart distribution. In addition, we collect all terms related to process noise covariance together to obtain the following.

$$p(\mathbf{x}_n, \mathbf{Q}_n|\mathbf{y}_{0:N}) = \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n})$$
$$\times \int \mathrm{St}\left(\mathbf{x}_{n+1}; \mathbf{A}\mathbf{x}_n, \frac{\mathbf{V}_{n+1|N}}{\upsilon_{n+1|N} - 2n_x}, \upsilon_{n+1|N} - 2n_x\right)$$
$$\times \int \mathcal{IW}(\mathbf{Q}_{n+1}; \upsilon_{n+1|N} + 1, \mathbf{V}_{n+1|N} + (\mathbf{x}_{n+1} - \mathbf{A}\mathbf{x}_n)(\mathbf{x}_{n+1} - \mathbf{A}\mathbf{x}_n)^{\mathrm{T}})$$
$$\times \frac{p(\mathbf{Q}_{n+1}|\mathbf{Q}_n)\mathcal{IW}(\mathbf{Q}_n; \upsilon_{n|n}, \mathbf{V}_{n|n})}{\mathcal{IW}(\mathbf{Q}_{n+1}; \upsilon_{n+1|n}, \mathbf{V}_{n+1|n})}\, \mathrm{d}\mathbf{Q}_{n+1}$$
$$\times \frac{\mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|N}, \mathbf{P}_{n+1|N})}{\mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|n}, \hat{\mathbf{P}}_{n+1|n})}\, \mathrm{d}\mathbf{x}_{n+1}. \tag{4.56}$$

The integral with respect to $\mathbf{Q}_{n+1}$ can be evaluated by using Beta-Bartlett smoothing expressions [79] as $\mathcal{IW}(\mathbf{Q}_n; \upsilon_{ns}, \mathbf{V}_{ns}(\mathbf{x}_n, \mathbf{x}_{n+1}))$ where

$$\mathbf{V}_{ns}(\mathbf{x}_n, \mathbf{x}_{n+1}) = \left((1-\gamma_Q)\mathbf{V}_{n|n}^{-1} + \gamma_Q\left(\mathbf{V}_{n+1|N} + (\mathbf{x}_{n+1} - \mathbf{A}\mathbf{x}_n)(\cdot)^{\mathrm{T}}\right)^{-1}\right)^{-1}, \tag{4.57a}$$

$$\upsilon_{ns} = (1-\gamma_Q)\upsilon_{n|n} + \gamma_Q(\upsilon_{n+1|N} + 1). \tag{4.57b}$$

Hence, we can express the distribution $p(\mathbf{x}_n, \mathbf{Q}_n|\mathbf{y}_{0:N})$ as

$$p(\mathbf{x}_n, \mathbf{Q}_n|\mathbf{y}_{0:N}) = \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n})$$
$$\times \int \mathrm{St}\left(\mathbf{x}_{n+1}; \mathbf{A}\mathbf{x}_n, \overline{\mathbf{Q}}_{n+1|N}, \upsilon_{n+1|N} - 2n_x\right)$$
$$\times \mathcal{IW}\left(\mathbf{Q}_n; \upsilon_{ns}, \mathbf{V}_{ns}(\mathbf{x}_n, \mathbf{x}_{n+1})\right) \frac{\mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|N}, \mathbf{P}_{n+1|N})}{\mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|n}, \hat{\mathbf{P}}_{n+1|n})}\, \mathrm{d}\mathbf{x}_{n+1}, \tag{4.58}$$

where

$$\overline{\mathbf{Q}}_{n+1|N} \triangleq \frac{\mathbf{V}_{n+1|N}}{\upsilon_{n+1|N} - 2n_x}. \tag{4.59}$$

As in the filtering part, we now approximate the $t$-distribution as a scale Gaussian mixture as follows:

$$\mathrm{St}\left(\mathbf{x}_{n+1}; \mathbf{A}\mathbf{x}_n, \overline{\mathbf{Q}}_{n+1|N}, v_{n+1|N} - 2n_x\right) \approx \sum_{i=1}^{I} \alpha_{si} \mathcal{N}\left(\mathbf{x}_{n+1}; \mathbf{A}\mathbf{x}_n, \frac{\overline{\mathbf{Q}}_{n+1|N}}{\beta_{si}}\right), \tag{4.60}$$

where $\{\beta_{si}\}_{i=1}^{I}$ denote the uniformly spaced grid points in the interval $[\max(0, \mu_{\mathcal{G}} - 3\sigma_{\mathcal{G}}),\ \mu_{\mathcal{G}} + 3\sigma_{\mathcal{G}}]$ where $\mu_{\mathcal{G}}$ and $\sigma_{\mathcal{G}}$ are the mean and standard deviation of the density $\mathcal{G}(\lambda; \nu/2, \nu/2)$ with $\nu = v_{n+1|N} - 2n_x$, respectively, and the weights $\{\alpha_{si}\}_{i=1}^{I}$ satisfy $\alpha_{si} \propto \mathcal{G}(\beta_{si}; \nu/2, \nu/2)$. Replacing the $t$-distribution in (4.58) with the approximate Gaussian mixture in (4.60) we get

$$\begin{aligned}
p(\mathbf{x}_n, \mathbf{Q}_n|\mathbf{y}_{0:N}) &= \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n}) \int \sum_{i=1}^{I} \alpha_{si} \mathcal{N}\left(\mathbf{x}_{n+1}; \mathbf{A}\mathbf{x}_n, \frac{\overline{\mathbf{Q}}_{n+1|N}}{\beta_{si}}\right) \\
&\quad \times \mathcal{IW}\left(\mathbf{Q}_n; v_{ns}, \mathbf{V}_{ns}(\mathbf{x}_n, \mathbf{x}_{n+1})\right) \frac{\mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|N}, \mathbf{P}_{n+1|N})}{\mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|n}, \hat{\mathbf{P}}_{n+1|n})}\, \mathrm{d}\mathbf{x}_{n+1} && \text{(4.61a)} \\
&= \sum_{i=1}^{I} \alpha_{si} \int \mathcal{IW}\left(\mathbf{Q}_n; v_{ns}, \mathbf{V}_{ns}(\mathbf{x}_n, \mathbf{x}_{n+1})\right) \frac{\mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|n}, \overline{\mathbf{P}}^{i}_{n+1|n})}{\mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|n}, \hat{\mathbf{P}}_{n+1|n})} \\
&\quad \times \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}^{i}_{n|n+1}(\mathbf{x}_{n+1}), \mathbf{P}^{i}_{n|n+1}) \mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|N}, \mathbf{P}_{n+1|N})\, \mathrm{d}\mathbf{x}_{n+1}, && \text{(4.61b)}
\end{aligned}$$

where

$$\begin{aligned}
\overline{\mathbf{P}}^{i}_{n+1|n} &\triangleq \mathbf{A}\mathbf{P}_{n|n}\mathbf{A}^{\mathrm{T}} + \frac{\overline{\mathbf{Q}}_{n+1|N}}{\beta_{si}}, && \text{(4.62a)} \\
\hat{\mathbf{x}}^{i}_{n|n+1}(\mathbf{x}_{n+1}) &\triangleq \hat{\mathbf{x}}_{n|n} + \mathbf{P}_{n|n}A^{\mathrm{T}}(\overline{\mathbf{P}}^{i}_{n+1|n})^{-1}(\mathbf{x}_{n+1} - \hat{\mathbf{x}}_{n+1|n}), && \text{(4.62b)} \\
\mathbf{P}^{i}_{n|n+1} &\triangleq \mathbf{P}_{n|n} - \mathbf{P}_{n|n}\mathbf{A}^{\mathrm{T}}(\overline{\mathbf{P}}^{i}_{n+1|n})^{-1}\mathbf{A}\mathbf{P}_{n|n}. && \text{(4.62c)}
\end{aligned}$$

We now approximate the ratio $\frac{\mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|n}, \overline{\mathbf{P}}^{i}_{n+1|n})}{\mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|n}, \hat{\mathbf{P}}_{n+1|n})}$ in (4.61b) as

$$\frac{\mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|n}, \overline{\mathbf{P}}^{i}_{n+1|n})}{\mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|n}, \hat{\mathbf{P}}_{n+1|n})} \approx \frac{\mathcal{N}(\hat{\mathbf{x}}_{n+1|N}; \hat{\mathbf{x}}_{n+1|n}, \overline{\mathbf{P}}^{i}_{n+1|n})}{\mathcal{N}(\hat{\mathbf{x}}_{n+1|N}; \hat{\mathbf{x}}_{n+1|n}, \hat{\mathbf{P}}_{n+1|n})}, \tag{4.63}$$

which gives

$$\begin{aligned}
p(\mathbf{x}_n, \mathbf{Q}_n|\mathbf{y}_{0:N}) &= \sum_{i=1}^{I} \alpha_i \int \mathcal{IW}\left(\mathbf{Q}_n; v_{ns}, \mathbf{V}_{ns}(\mathbf{x}_n, \mathbf{x}_{n+1})\right) \\
&\quad \times \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}^{i}_{n|n+1}(\mathbf{x}_{n+1}), \mathbf{P}^{i}_{n|n+1}) \mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|N}, \mathbf{P}_{n+1|N})\, \mathrm{d}\mathbf{x}_{n+1}, && \text{(4.64)}
\end{aligned}$$

where

$$\alpha_i \propto \alpha_{si} \frac{\mathcal{N}(\hat{\mathbf{x}}_{n+1|N}; \hat{\mathbf{x}}_{n+1|n}, \overline{\mathbf{P}}^i_{n+1|n})}{\mathcal{N}(\hat{\mathbf{x}}_{n+1|N}; \hat{\mathbf{x}}_{n+1|n}, \hat{\mathbf{P}}_{n+1|n})}. \tag{4.65}$$

We now approximate $p(\mathbf{x}_n, \mathbf{Q}_n|\mathbf{y}_{0:N})$ as $p(\mathbf{x}_n, \mathbf{Q}_n|\mathbf{y}_{0:N}) \approx p(\mathbf{x}_n|\mathbf{y}_{0:N})p(\mathbf{Q}_n|\mathbf{y}_{0:N})$ where $p(\mathbf{x}_n|\mathbf{y}_{0:N})$ and $p(\mathbf{Q}_n|\mathbf{y}_{0:N})$ are found by marginalization of (4.64).

$$p(\mathbf{x}_n|\mathbf{y}_{0:N}) = \int p(\mathbf{x}_n, \mathbf{Q}_n|\mathbf{y}_{0:N})\, \mathrm{d}\mathbf{Q}_n \tag{4.66a}$$

$$= \sum_{i=1}^{I} \alpha_i \int \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}^i_{n|n+1}(\mathbf{x}_{n+1}), \mathbf{P}^i_{n|n+1}) \mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|N}, \mathbf{P}_{n+1|N})\, \mathrm{d}\mathbf{x}_{n+1} \tag{4.66b}$$

$$= \sum_{i=1}^{I} \alpha_i \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}^i_{n|N}, \mathbf{P}^i_{n|N}) \tag{4.66c}$$

$$\approx \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|N}, \mathbf{P}_{n|N}), \tag{4.66d}$$

where

$$\hat{\mathbf{x}}^i_{n|N} = \hat{\mathbf{x}}^i_{n|n+1}(\hat{\mathbf{x}}_{n+1|N}) = \hat{\mathbf{x}}_{n|n} + \mathbf{P}_{n|n}\mathbf{A}^{\mathrm{T}}(\overline{\mathbf{P}}^i_{n+1|n})^{-1}(\hat{\mathbf{x}}_{n+1|N} - \hat{\mathbf{x}}_{n+1|n}), \tag{4.67a}$$

$$\mathbf{P}^i_{n|N} = \mathbf{P}_{n|n}\mathbf{A}^{\mathrm{T}}(\overline{\mathbf{P}}^i_{n+1|n})^{-1}\mathbf{P}_{n+1|N}(\overline{\mathbf{P}}^i_{n+1|n})^{-1}\mathbf{A}\mathbf{P}_{n|n} + \mathbf{P}^i_{n|n+1} \tag{4.67b}$$

$$= \mathbf{P}_{n|n}\mathbf{A}^{\mathrm{T}}(\overline{\mathbf{P}}^i_{n+1|n})^{-1}\mathbf{P}_{n+1|N}(\overline{\mathbf{P}}^i_{n+1|n})^{-1}\mathbf{A}\mathbf{P}_{n|n} + \mathbf{P}_{n|n} - \mathbf{P}_{n|n}\mathbf{A}^{\mathrm{T}}(\overline{\mathbf{P}}^i_{n+1|n})^{-1}\mathbf{A}\mathbf{P}_{n|n} \tag{4.67c}$$

$$= \mathbf{P}_{n|n} - \mathbf{P}_{n|n}A^{\mathrm{T}}(\overline{\mathbf{P}}^i_{n+1|n})^{-1}(\overline{\mathbf{P}}^i_{n+1|n} - \mathbf{P}_{n+1|N})(\overline{\mathbf{P}}^i_{n+1|n})^{-1}\mathbf{A}\mathbf{P}_{n|n}, \tag{4.67d}$$

$$\hat{\mathbf{x}}_{n|N} = \sum_{i=1}^{I} \alpha_i \hat{\mathbf{x}}^i_{n|N}, \tag{4.67e}$$

$$\mathbf{P}_{n|N} = \sum_{i=1}^{I} \alpha_i \left( \hat{\mathbf{P}}^i_{n|N} + (\hat{\mathbf{x}}^i_{n|N} - \hat{\mathbf{x}}_{n|N})(\hat{\mathbf{x}}^i_{n|N} - \hat{\mathbf{x}}_{n|N})^{\mathrm{T}} \right). \tag{4.67f}$$

We can calculate $p(\mathbf{Q}_n|\mathbf{y}_{0:N})$ by marginalizing $p(\mathbf{x}_n, \mathbf{Q}_n|\mathbf{y}_{0:N})$ in (4.64) with respect to $\mathbf{x}_n$ as

$$p(\mathbf{Q}_n|\mathbf{y}_{0:N}) = \sum_{i=1}^{I} \alpha_i \int\int \mathcal{IW}(\mathbf{Q}_n; v_{ns}, \mathbf{V}_{ns}(\mathbf{x}_n, \mathbf{x}_{n+1})) \times \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}^i_{n|n+1}(\mathbf{x}_{n+1}), \mathbf{P}^i_{n|n+1}) \mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|N}, \mathbf{P}_{n+1|N})\, \mathrm{d}\mathbf{x}_{n+1}\, \mathrm{d}\mathbf{x}_n. \tag{4.68}$$

We now represent the multiplication of the Gaussian distributions in (4.68) as a joint distribution as follows:

$$\mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}^i_{n|n+1}(\mathbf{x}_{n+1}), \mathbf{P}^i_{n|n+1}) \mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|N}, \mathbf{P}_{n+1|N})$$

$$= \mathcal{N}\left(\begin{bmatrix}\mathbf{x}_{n+1}\\ \mathbf{x}_n\end{bmatrix}; \begin{bmatrix}\hat{\mathbf{x}}_{n+1|N}\\ \hat{\mathbf{x}}^i_{n|n+1}(\hat{\mathbf{x}}_{n+1|N})\end{bmatrix}, \begin{bmatrix}\mathbf{P}_{n+1|N} & \mathbf{P}_{n+1|N}\mathbf{K}_i^{\mathrm{T}}\\ \mathbf{K}_i\mathbf{P}_{n+1|N} & \mathbf{P}^i_{n|N}\end{bmatrix}\right) \tag{4.69a}$$

$$= \mathcal{N}\left(\mathbf{z}_n; \hat{\mathbf{z}}_i, \mathbf{Z}_i\right), \tag{4.69b}$$

where $\mathbf{K}_i$ is

$$\mathbf{K}_i = \mathbf{P}_{n|n}\mathbf{A}^{\mathrm{T}}(\overline{\mathbf{P}}^i_{n+1|n})^{-1}. \tag{4.70}$$

Defining the random vector $\mathbf{w}_n \triangleq \mathbf{x}_{n+1} - \mathbf{A}\mathbf{x}_n$, we can express $p(\mathbf{Q}_n|\mathbf{y}_{0:N})$ as follows:

$$p(\mathbf{Q}_n|\mathbf{y}_{0:N}) = \sum_{i=1}^{I}\alpha_i \int \mathcal{N}(\mathbf{w}_n; \hat{\mathbf{w}}_i, \hat{\mathbf{W}}_i) \times \mathcal{IW}\left(\mathbf{Q}_n; \upsilon_{ns}, \left((1-\gamma_Q)\mathbf{V}^{-1}_{n|n} + \gamma_Q(\mathbf{V}_{n+1|N} + \mathbf{w}_n\mathbf{w}_n^{\mathrm{T}})^{-1}\right)^{-1}\right) \mathrm{d}\mathbf{w}_n \tag{4.71a}$$

$$= \int \left(\sum_{i=1}^{I}\alpha_i\mathcal{N}(\mathbf{w}_n; \hat{\mathbf{w}}_i, \hat{\mathbf{W}}_i)\right) \times \mathcal{IW}\left(\mathbf{Q}_n; \upsilon_{ns}, \left((1-\gamma_Q)\mathbf{V}^{-1}_{n|n} + \gamma_Q(\mathbf{V}_{n+1|N} + \mathbf{w}_n\mathbf{w}_n^{\mathrm{T}})^{-1}\right)^{-1}\right) \mathrm{d}\mathbf{w}_n \tag{4.71b}$$

$$\approx \mathcal{IW}(\mathbf{Q}_n; \upsilon_{n|N}, \mathbf{V}_{n|N}), \tag{4.71c}$$

where $\hat{\mathbf{w}}_i$ and $\hat{\mathbf{W}}_i$ are given as

$$\hat{\mathbf{w}}_i = \begin{bmatrix}\mathcal{I} & -\mathbf{A}\end{bmatrix}\hat{\mathbf{z}}_i, \tag{4.72a}$$

$$\hat{\mathbf{W}}_i = \begin{bmatrix}\mathcal{I} & -\mathbf{A}\end{bmatrix}\mathbf{Z}_i\begin{bmatrix}\mathcal{I}\\ -\mathbf{A}^{\mathrm{T}}\end{bmatrix}. \tag{4.72b}$$

In (4.71), we use moment matching as in Section 4.10.2. The smoothed parameters $\upsilon_{n|N}$ and $\mathbf{V}_{n|N}$ are given as

$$\upsilon_{n|N} = 2n_x + 2 + \frac{2(\upsilon_{ns} - 2n_x - 2)}{2\alpha + (1-\alpha)(\upsilon_{ns} - 2n_x - 2)}, \tag{4.73a}$$

$$\mathbf{V}_{n|N} = \frac{\upsilon_{n|N} - 2n_x - 2}{\upsilon_{ns} - 2n_x - 2}\,\mathrm{E}\left[\mathbf{V}(\mathbf{w}_n)\right], \tag{4.73b}$$

where

$$\alpha = \frac{\sum_{m=1}^{n_x}\mathrm{E}^2\left[\mathbf{V}_{mm}(\mathbf{w}_n)\right]}{\sum_{m=1}^{n_x}\mathrm{E}\left[\mathbf{V}^2_{mm}(\mathbf{w}_n)\right]}, \tag{4.74a}$$

$$\mathbf{V}(\mathbf{w}_n) = \left((1-\gamma_Q)\mathbf{V}^{-1}_{n|n} + \gamma_Q(\mathbf{V}_{n+1|N} + \mathbf{w}_n\mathbf{w}_n^{\mathrm{T}})^{-1}\right)^{-1}. \tag{4.74b}$$

The expectations $\mathrm{E}[\mathbf{V}_{mm}(\mathbf{w}_n)]$ and $\mathrm{E}[\mathbf{V}^2_{mm}(\mathbf{w}_n)]$ required above are to be taken with respect to the Gaussian mixture

$$p(\mathbf{w}_n) \triangleq \sum_{i=1}^{I} \alpha_i \mathcal{N}(\mathbf{w}_n; \hat{\mathbf{w}}_i, \hat{\mathbf{W}}_i). \tag{4.75}$$

Approximate analytical expressions to calculate $\mathrm{E}[\mathbf{V}_{mm}(\mathbf{w}_n)]$ and $\mathrm{E}[\mathbf{V}^2_{mm}(\mathbf{w}_n)]$ are given in Section 4.10.4. The algorithm related to this part is given in Algorithm 4.

**Algorithm 4** Smoothing with Unknown PNC

1: **Inputs:** $\mathbf{A}, I, \gamma_Q, \hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n}, \hat{\mathbf{x}}_{n+1|n}, \hat{\mathbf{x}}_{n+1|N}, \mathbf{P}_{n+1|N}, \upsilon_{n|n}, \mathbf{V}_{n|n}, \upsilon_{n+1|N}, \mathbf{V}_{n+1|N}$

2: Calculate the grid-points $\{\beta_{si}\}_{i=1}^{I}$ and their weights $\{\alpha_{si}\}_{i=1}^{I}$ from Gamma distribution with parameters $\alpha = \beta = (\upsilon_{n+1|N} - 2n_x)/2$.

3: Calculate $\{\hat{\mathbf{x}}^i_{n|N}, \mathbf{P}^i_{n|N}\}_{i=1}^{I}$ using (4.67a) and (4.67d), respectively.

4: Merge $\{\hat{\mathbf{x}}^i_{n|N}, \hat{\mathbf{P}}^i_{n|N}\}_{i=1}^{I}$ using (4.67e) and (4.67f) to obtain $\hat{\mathbf{x}}_{n|N}$ and $\mathbf{P}_{n|N}$, respectively.

5: Calculate moments $\mathrm{E}[\mathbf{V}(\mathbf{w}_n)]$, $\mathrm{E}[\mathbf{V}_{mm}(\mathbf{w}_n)]$ and $\mathrm{E}[\mathbf{V}^2_{mm}(\mathbf{w}_n)]$ using (4.121), (4.123) and (4.124), respectively.

6: Calculate $\upsilon_{n|N}$ and $\mathbf{V}_{n|N}$ using (4.73).

7: **Outputs:** $\hat{\mathbf{x}}_{n|N}, \mathbf{P}_{n|N}, \upsilon_{n|N}, \mathbf{V}_{n|N}$

## 4.6 Bayesian Smoothing with Unknown MNC

The smoothed posterior $p(\mathbf{x}_n, \mathbf{R}_n|\mathbf{y}_{0:N})$ is given as

$$p(\mathbf{x}_n, \mathbf{R}_n|\mathbf{y}_{0:N}) = \int \frac{p(\mathbf{x}_{n+1}, \mathbf{R}_{n+1}|\mathbf{x}_n, \mathbf{R}_n) p(\mathbf{x}_n, \mathbf{R}_n|\mathbf{y}_{0:n})}{p(\mathbf{x}_{n+1}, \mathbf{R}_{n+1}|\mathbf{y}_{0:n})} \times p(\mathbf{x}_{n+1}, \mathbf{R}_{n+1}|\mathbf{y}_{0:N})\, \mathrm{d}\mathbf{x}_{n+1}\, \mathrm{d}\mathbf{R}_{n+1}, \tag{4.76a}$$

$$= \int \frac{\mathcal{N}(\mathbf{x}_{n+1}; \mathbf{A}\mathbf{x}_n, \mathbf{Q}) \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n})}{\mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|n}, \mathbf{P}_{n+1|n})} \mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|N}, \mathbf{P}_{n+1|N})\, \mathrm{d}\mathbf{x}_{n+1} \times \int \frac{p(\mathbf{R}_{n+1}|\mathbf{R}_n) \mathcal{IW}(\mathbf{R}_n; w_{n|n}, \mathbf{W}_{n|n})}{\mathcal{IW}(\mathbf{R}_{n+1}; w_{n+1|n}, \mathbf{W}_{n+1|n})} \mathcal{IW}(\mathbf{R}_{n+1}; w_{n+1|N}, \mathbf{W}_{n+1|N})\, \mathrm{d}\mathbf{R}_{n+1}, \tag{4.76b}$$

$$= \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|N}, \mathbf{P}_{n|N}) \mathcal{IW}(\mathbf{R}_n; w_{n|N}, \mathbf{W}_{n|N}), \tag{4.76c}$$

where

$$\hat{\mathbf{x}}_{n|N} = \hat{\mathbf{x}}_{n|n} + \mathbf{G_n}[\hat{\mathbf{x}}_{n+1|N} - \hat{\mathbf{x}}_{n+1|n}], \tag{4.77a}$$

$$\mathbf{P}_{n|N} = \mathbf{P}_{n|n} + \mathbf{G_n}[\mathbf{P}_{n+1|N} - \mathbf{P}_{n+1|n}]\mathbf{G_n}^{\mathrm{T}}, \tag{4.77b}$$

$$\mathbf{G_n} = \mathbf{P}_{n|n}\mathbf{A}^{\mathrm{T}}\mathbf{P}_{n+1|n}^{-1}, \tag{4.77c}$$

$$w_{n|N} = (1-\gamma_R)w_{n|n} + \gamma_R w_{n+1|N}, \tag{4.77d}$$

$$\mathbf{W}_{n|N} = \left((1-\gamma_R)\mathbf{W}_{n|n}^{-1} + \gamma_R \mathbf{W}_{n+1|N}^{-1}\right)^{-1}. \tag{4.77e}$$

## 4.7 Simulation Results

We consider the two-dimensional target tracking problem given in [79] where a single target with its state composed of $x, y$ positions and velocities, i.e., $\mathbf{x}_n \triangleq [x_n, \dot{x}_n, y_n, \dot{y}_n]^{\mathrm{T}}$, moves with nearly constant velocity. The state space model parameters are given as [8]

$$\mathbf{A} = \begin{bmatrix} \bar{\mathbf{A}} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{A}} \end{bmatrix}, \qquad \bar{\mathbf{A}} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix},$$
$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{4.78}$$

where $\Delta t = 1s$. The initial state is chosen as Gaussian with mean $\mathbf{x}_0 = [0, 5, 0, 5]^{\mathrm{T}}$ and covariance $\mathbf{\Sigma}_0 = \mathrm{diag}([30^2, 30^2, 30^2, 30^2])$.

### 4.7.1 Bayesian Filtering and Smoothing with Unknown PNC

In this section, we investigate the case of unknown PNC. We assume that MNC is exactly known and it is given as

$$\mathbf{R}_n = \sigma_e^2 \begin{bmatrix} 5 & 1 \\ 1 & 5 \end{bmatrix}, \tag{4.79}$$

for $n = 0, 1, \ldots, N$ where $\sigma_e^2 = 2m^2$. For the PNC we consider two cases:

- Constant PNC: In this case the PNC is constant in time and given as

$$\mathbf{Q}_n = \begin{bmatrix} \bar{\mathbf{Q}} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{Q}} \end{bmatrix}, \qquad \bar{\mathbf{Q}} = \sigma_v^2 \begin{bmatrix} \Delta t^3/3 & \Delta t^2/2 \\ \Delta t^2/2 & \Delta t \end{bmatrix} \tag{4.80}$$

  for $n = 0, 1, \ldots, N$ where $\sigma_v^2 = 27m^2/s^3$.

- Time-varying PNC: The PNC $\mathbf{Q}_n$ changes in time as follows.

$$\mathbf{Q}_n = \left(\frac{2}{3} + \frac{1}{3}\cos\left(\frac{4\pi n}{N}\right)\right)\begin{bmatrix}\bar{\mathbf{Q}} & \mathbf{0}\\ \mathbf{0} & \bar{\mathbf{Q}}\end{bmatrix}, \quad \bar{\mathbf{Q}} = \sigma_v^2\begin{bmatrix}\Delta t^3/3 & \Delta t^2/2\\ \Delta t^2/2 & \Delta t\end{bmatrix} \tag{4.81}$$

  for $n = 0, 1, \ldots, N$ where $\sigma_v^2 = 27m^2/s^3$.

The scenario length is chosen as $N = 4000$. For the purpose of comparison, the following smoothers are implemented.

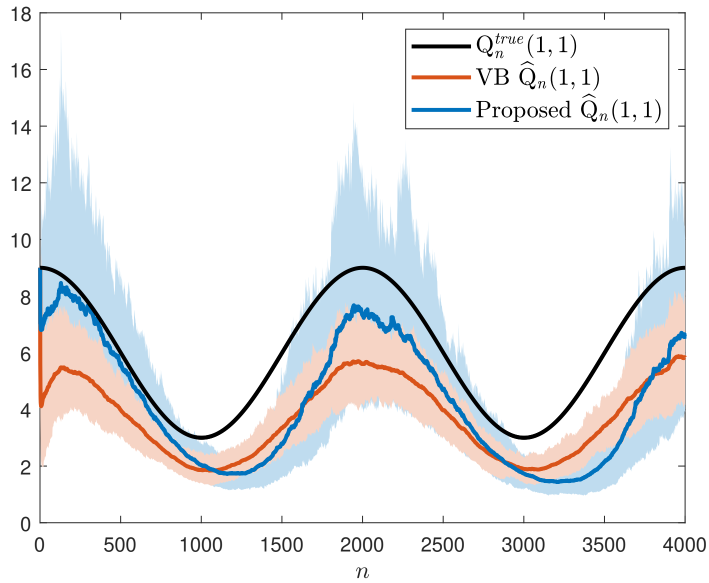
- **Kalman smoother (RTS):** A clairvoyant Rauch-Tung-Striebel (RTS) smoother using the true PNCs.
- **VB:** The explicit variational Bayes smoother proposed in [79]. The number of iterations is set to $50$ as in [79]. The covariance discount factors are chosen as $\gamma_Q = 0.98$ and $\gamma_Q = 1$ for time-varying and constant process noise covariance scenarios, respectively.
- **Proposed:** The proposed smoother in this work. The number of components used to represent Student's *t*-distribution is set to $10$. The covariance discount factors are chosen as $\gamma_Q = 0.98$ and $\gamma_Q = 1$ for time-varying and constant process noise covariance scenarios, respectively. The initial PNC distribution parameters are selected as $v_0 = 2n_x + 3$ and $\mathbf{V}_0 = (v_0 - 2n_x - 2)\mathbf{Q}_0$, which results in the initial mean (of the PNC) being equal to its true value.

Note that the smoothers mentioned above perform filtering as well. Thus, we compare the performances of both filters and smoothers. A total of $100$ Monte Carlo (MC) runs are made by changing the state and measurement realizations in each run. We investigate the performance in terms of root mean square error (RMSE) of position and velocity estimates (over the MC runs). While comparing the performances, the average RMSEs (ARMSE) over time are used. We also investigate the biases in the covariance estimates. To get the bias information for each time index, we first compute the average of covariance estimates over Monte Carlo runs. The true covariance values are then subtracted from the averaged covariance estimates. We use the mean of absolute bias values obtained at each time index to compare the bias information.

When we run our algorithm and the VB algorithm for the time-varying PNC scenario, we obtain the results of filtered and smoothed PNC estimates as shown in Figures 4.1 and 4.2, respectively. These figures include mean estimates and uncertainties of the estimates. The mean estimates for the filtered and smoothed estimates are found as $\mathbf{V}_{n|n}/(v_{n|n} - 2n_x - 2)$ and $\mathbf{V}_{n|N}/(v_{n|N} - 2n_x - 2)$, respectively. The results show that the variances of the covariance estimates are higher for the proposed filtering and smoothing solutions.

Average RMSE values for the filtering and smoothing solutions for the time-varying PNC scenario are given in Table 4.1. The results for the constant PNC is given in Table 4.2. Table 4.1 and 4.2 show that the proposed algorithm and the VB method provide similar performance and their performance is close to that of RTS.

There is bias in the covariance estimates as seen in Figures 4.1 and 4.2. The bias values $\widehat{\mathbf{Q}}^b(1,1)$, $\widehat{\mathbf{Q}}^b(1,2)$ and $\widehat{\mathbf{Q}}^b(2,2)$ for the filtered and smoothed PNC estimates are given in Table 4.3. The bias of the proposed smoothing method is greater than the bias of the VB method for the time-varying noise covariance simulation. In addition, the proposed method yields less bias than the VB method for the constant noise covariance case. These bias values affect the accuracy of the state estimates as



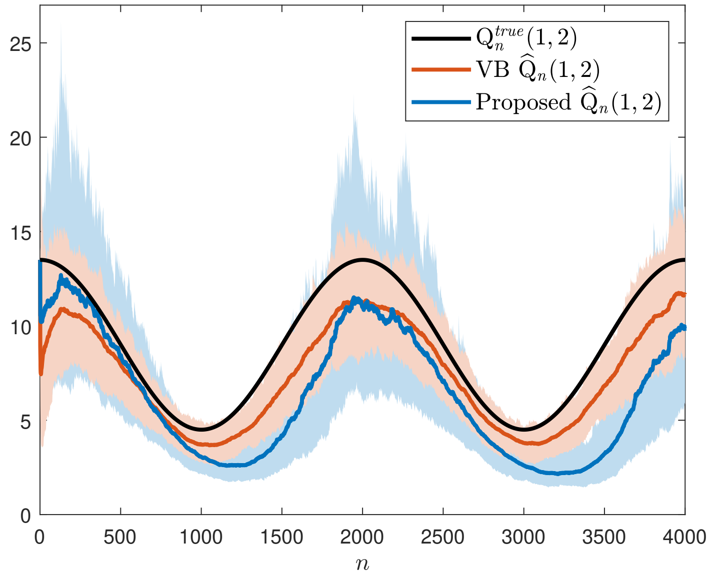

(a) Filtered $\widehat{\mathbf{Q}}_n(1,1)$

Table 4.1: Position and velocity ARMSE of the filters and smoothers for the time-varying PNC scenario.

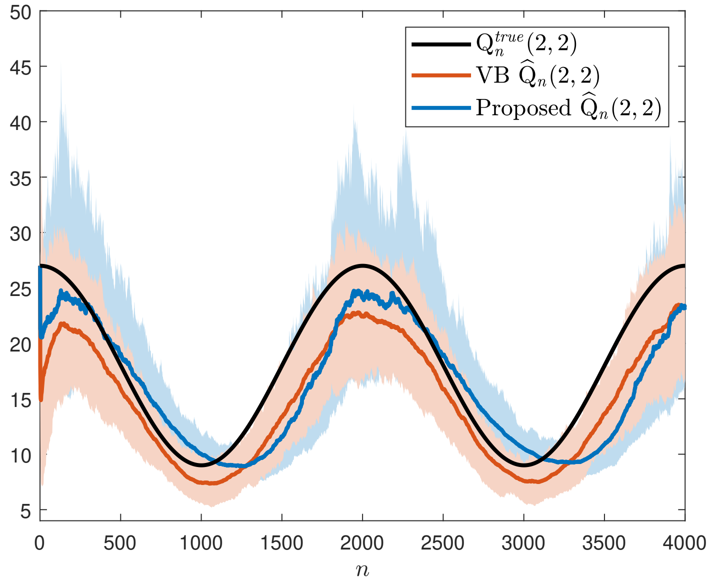
| Filter/ Smoother | Filt. Pos. ARMSE (m) | Filt. Vel. ARMSE (m/s) | Smoo. Pos. ARMSE (m) | Smoo. Vel. ARMSE (m/s) |
|---|---|---|---|---|
| RTS | 3.992 | 5.469 | 2.831 | 3.281 |
| VB | 4.000 | 5.470 | 2.863 | 3.293 |
| Proposed | 4.005 | 5.507 | 2.886 | 3.419 |

Table 4.2: Position and velocity ARMSE of the filters and smoothers for the constant PNC scenario.

| Filter/ Smoother | Filt. Pos. ARMSE (m) | Filt. Vel. ARMSE (m/s) | Smoo. Pos. ARMSE (m) | Smoo. Vel. ARMSE (m/s) |
|---|---|---|---|---|
| RTS | 4.085 | 6.387 | 2.997 | 3.892 |
| VB | 4.086 | 6.393 | 3.002 | 3.901 |
| Proposed | 4.087 | 6.393 | 3.000 | 3.897 |




(b) Filtered $\widehat{\mathbf{Q}}_n(1,2)$

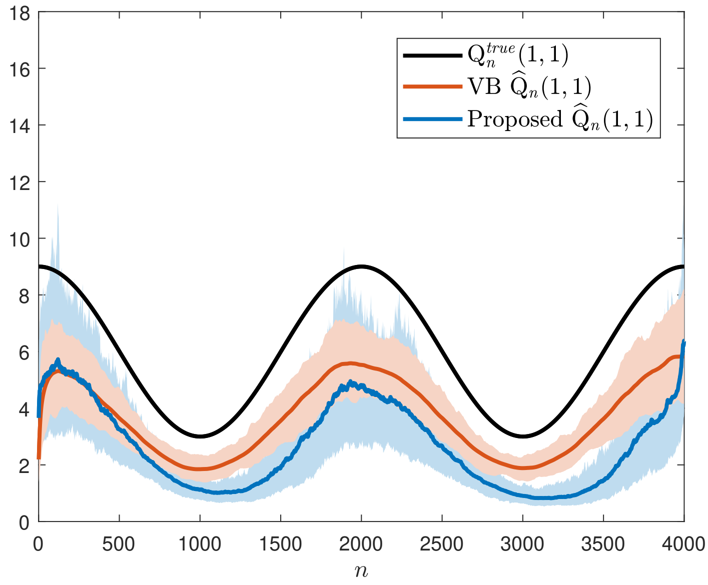



(c) Filtered $\widehat{\mathbf{Q}}_n(2,2)$

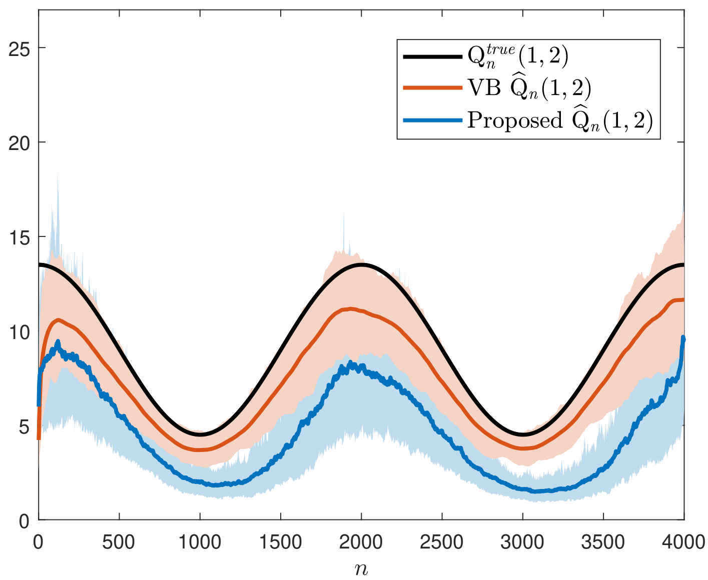
Figure 4.1: Filtered mean estimates and uncertainties of some entries of the PNC for the proposed method and VB in the time-varying PNC scenario. The estimate uncertainties are shown by the shaded regions representing the interval between the 5th and 95th percentiles over the MC runs.

seen in Tables 4.1 and 4.2. Finally, the bias values for time-varying noise covariance cases are greater than those for constant noise covariance cases.

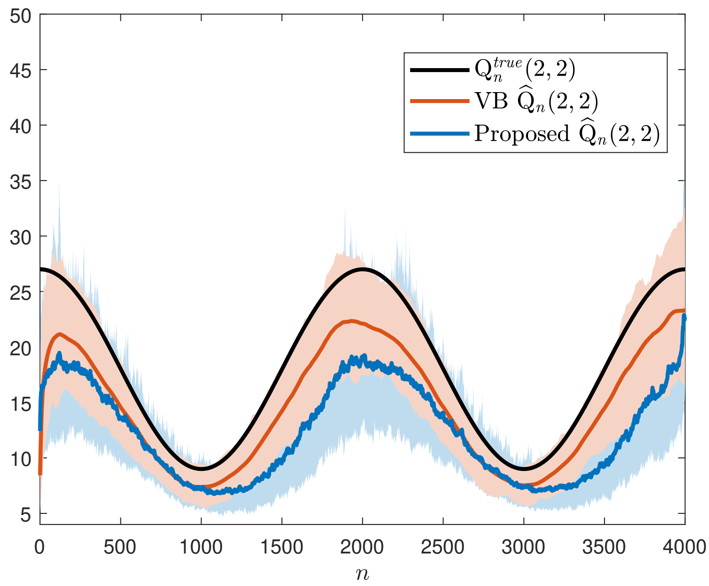
The results in this part show that the VB and the proposed solution provide similar results. The superiority of the proposed algorithm is the computation time. The computation times per MC run of the smoothers are given in Table 4.4.




(a) Smoothed $\widehat{\mathbf{Q}}_n(1,1)$




(b) Smoothed $\widehat{\mathbf{Q}}_n(1,2)$




(c) Smoothed $\widehat{\mathbf{Q}}_n(2,2)$

Figure 4.2: Smoothed mean estimates and uncertainties of some entries of the PNC for the proposed method and VB in the time-varying PNC scenario. The estimate uncertainties are shown by the shaded regions representing the interval between the 5th and 95th percentiles over the MC runs.

### 4.7.2 Bayesian Filtering and Smoothing with Unknown MNC

In this section, we investigate the case of unknown MNC. We assume that PNC is exactly known and it is given as

$$\mathbf{Q}_n = \begin{bmatrix} \bar{\mathbf{Q}} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{Q}} \end{bmatrix}, \qquad \bar{\mathbf{Q}} = \sigma_v^2 \begin{bmatrix} \Delta t^3/3 & \Delta t^2/2 \\ \Delta t^2/2 & \Delta t \end{bmatrix} \tag{4.82}$$

for $n = 0, 1, \ldots, N$ where $\sigma_v^2 = 27m^2/s^3$. For the MNC we consider two cases:

- Constant MNC: In this case the MNC is constant in time and given as

$$\mathbf{R}_n = \sigma_e^2 \begin{bmatrix} 5 & 1 \\ 1 & 5 \end{bmatrix} \text{ for } n = 0, 1, \ldots, N, \tag{4.83}$$

  where $\sigma_e^2 = 2m^2$.

Table 4.3: Bias values for the filtered and smoothed PNC estimates for the time-varying (TV) and constant (C) PNC scenarios.

| Filter/ Smoother | Filter $\widehat{\mathbf{Q}}^b(1,1)$ $(m^2)$ | Filter $\widehat{\mathbf{Q}}^b(1,2)$ $(m^2/s)$ | Filter $\widehat{\mathbf{Q}}^b(2,2)$ $(m^2/s^2)$ | Smoother $\widehat{\mathbf{Q}}^b(1,1)$ $(m^2)$ | Smoother $\widehat{\mathbf{Q}}^b(1,2)$ $(m^2/s)$ | Smoother $\widehat{\mathbf{Q}}^b(2,2)$ $(m^2/s^2)$ |
|---|---|---|---|---|---|---|
| VB (TV) | 2.290 | 1.596 | 3.191 | 2.337 | 1.686 | 3.372 |
| Proposed (TV) | 1.791 | 2.687 | 3.014 | 3.357 | 4.464 | 5.672 |
| VB (C) | 1.561 | 0.577 | 1.157 | 1.561 | 0.581 | 1.165 |
| Proposed (C) | 0.113 | 0.169 | 0.489 | 0.328 | 0.391 | 0.220 |

Table 4.4: Computation times (per MC run) of the smoothers for the time-varying PNC scenario.

| Smoother | Computation Times (s) |
|---|---|
| RTS | 0.91 |
| VB | 30.39 |
| Proposed | 14.93 |

- Time-varying MNC: The MNC $\mathbf{R}_n$ changes in time as follows.

$$\mathbf{R}_n = \sigma_e^2 \left(2 - \cos\left(\frac{4\pi n}{N}\right)\right) \begin{bmatrix} 5 & 1 \\ 1 & 5 \end{bmatrix}, \tag{4.84}$$

  for $n = 0, 1, \ldots, N$ where $\sigma_e^2 = 2m^2$,

The scenario length is chosen as $N = 4000$. For the purpose of comparison, the following smoothers are implemented.

- **Kalman smoother (RTS):** A clairvoyant Rauch-Tung-Striebel (RTS) smoother using the true MNCs.

- **RS-IID:** The implicit smoother proposed by Agamennoni et al. in [78]. The number of iterations is set to $50$. The covariance discount factor is chosen as $\gamma_R = 0.98$ for both time-varying and constant measurement noise covariances.

- **VB:** The explicit variational Bayes smoother proposed in [79]. The number of iterations is set to $50$ as in [79]. The covariance discount factors are chosen as $\gamma_R = 0.98$ and $\gamma_R = 1$ for time-varying and constant measurement noise covariance scenarios, respectively.

- **Proposed:** The proposed smoother in this work. The number of components used to represent Student's *t*-distribution is set to $10$. The covariance discount factors are chosen as $\gamma_R = 0.98$ and $\gamma_R = 1$ for time-varying and constant measurement noise covariance scenarios, respectively. The initial MNC distribution parameters are selected as $w_0 = 2n_y + 3$ and $\mathbf{W}_0 = (w_0 - 2n_y - 2)\mathbf{R}_0$, which results in the initial mean (of the MNC) being equal to its true value.

A total of $100$ Monte Carlo (MC) runs are made by changing the state and measurement realizations in each run. When we run our algorithm and the VB algorithm, we obtain the results of filtered MNC estimates and smoothed MNC estimates as shown in Figures 4.3 and 4.4. These figures include mean estimates and uncertainties of the estimates. The mean estimates for the filtered and smoothed estimates are found as $\mathbf{W}_{n|n}/(w_{n|n} - 2n_y - 2)$ and $\mathbf{W}_{n|N}/(w_{n|N} - 2n_y - 2)$, respectively. The VB and the proposed method provide similar mean estimates while the variances of the estimates are slightly greater for the proposed method.

The average RMSE values for time-varying noise covariance are given in Table 4.5. In the last simulation, we investigate constant MNC. The ARMSE performances of the filters and smoothers are given in Table 4.6. Similar to the unknown PNC simulations, Table 4.5 and 4.6 reveal that the proposed algorithm and the VB method show similar performance. Proposed algorithm outperforms the RS-IID method. Furthermore, the performance of the proposed algorithm is close to the performance of the RTS algorithm which knows the noise covariances exactly.

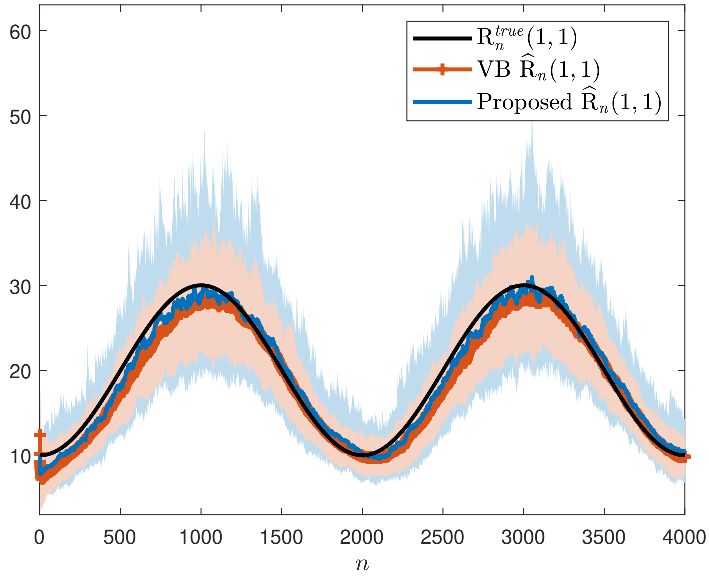
There is bias in the covariance estimates as seen in Figures 4.3 and 4.4. In the figures, it is seen that MNC is estimated with less bias compared to PNC estimations in pre-

Table 4.5: Position and velocity ARMSE of the filters and smoothers for the time-varying MNC scenario.

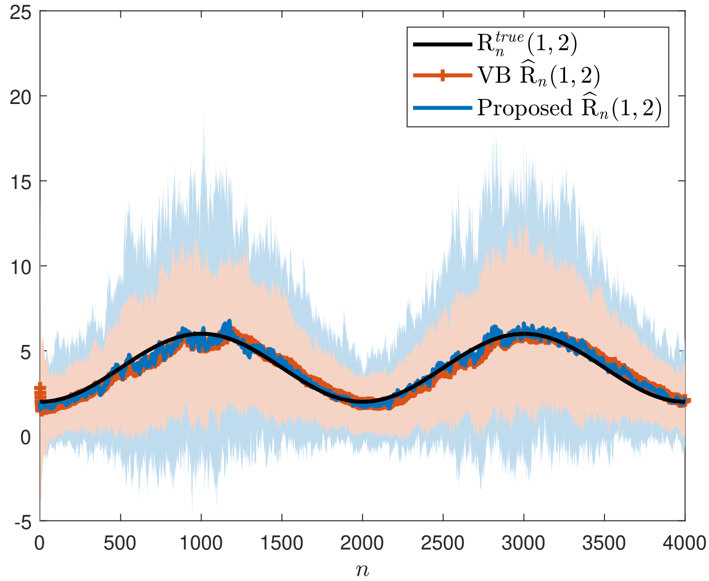
| Filter/ Smoother | Filt. Pos. ARMSE (m) | Filt. Vel. ARMSE (m/s) | Smoo. Pos. ARMSE (m) | Smoo. Vel. ARMSE (m/s) |
|---|---|---|---|---|
| RTS | 5.490 | 7.040 | 3.833 | 4.194 |
| VB | 5.485 | 7.043 | 3.839 | 4.201 |
| RS-IID | 5.526 | 7.182 | 3.882 | 4.254 |
| Proposed | 5.506 | 7.085 | 3.856 | 4.211 |

Table 4.6: Position and velocity ARMSE of the filters and smoothers for the constant MNC scenario.

| Filter/ Smoother | Filt. Pos. ARMSE (m) | Filt. Vel. ARMSE (m/s) | Smoo. Pos. ARMSE (m) | Smoo. Vel. ARMSE (m/s) |
|---|---|---|---|---|
| RTS | 4.080 | 6.387 | 2.991 | 3.890 |
| VB | 4.086 | 6.388 | 2.997 | 3.892 |
| RS-IID | 4.103 | 6.468 | 3.024 | 3.928 |
| Proposed | 4.087 | 6.391 | 2.999 | 3.894 |

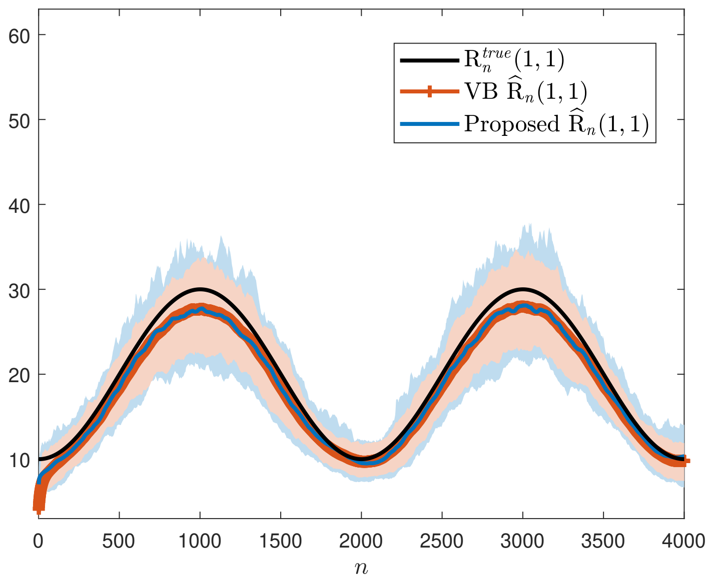



(a) Filtered $\widehat{\mathbf{R}}_n(1,1)$

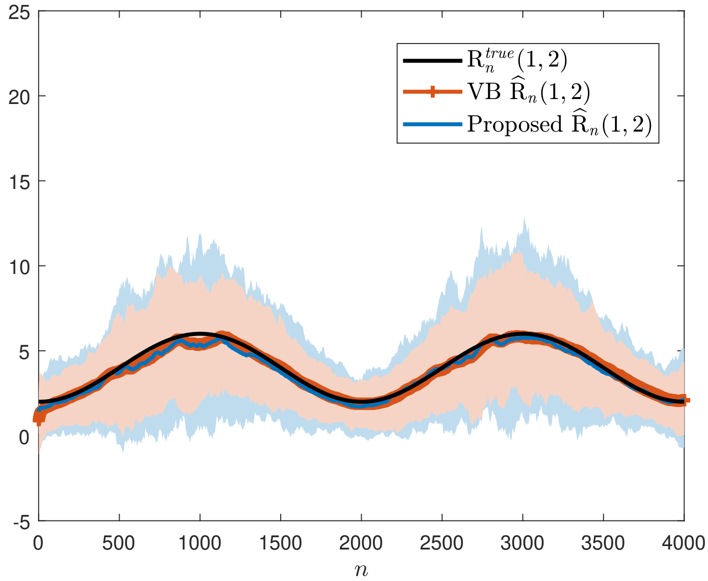



(b) Filtered $\widehat{\mathbf{R}}_n(1,2)$

Figure 4.3: Filtered mean estimates and uncertainties of some entries of the MNC for the proposed method and VB in the time-varying MNC scenario. The estimate uncertainties are shown by the shaded regions representing the interval between the 5th and 95th percentiles over the MC runs.




(a) Smoothed $\widehat{\mathbf{R}}_n(1,1)$




(b) Smoothed $\widehat{\mathbf{R}}_n(1,2)$

Figure 4.4: Smoothed mean estimates and uncertainties of some entries of the MNC for the proposed method and VB in the time-varying MNC scenario. The estimate uncertainties are shown by the shaded regions representing the interval between the 5th and 95th percentiles over the MC runs.

Table 4.7: Bias values for the filtered and smoothed MNC estimates for the time-varying (TV) and constant (C) MNC scenarios.

| Filter/ Smoother | Filter $\widehat{\mathbf{R}}^b(1,1)$ $(m^2)$ | Filter $\widehat{\mathbf{R}}^b(1,2)$ $(m^2)$ | Smoother $\widehat{\mathbf{R}}^b(1,1)$ $(m^2)$ | Smoother $\widehat{\mathbf{R}}^b(1,2)$ $(m^2)$ |
|---|---|---|---|---|
| VB (TV) | 1.347 | 0.313 | 1.389 | 0.163 |
| RS-IID (TV) | 6.293 | 1.029 | 6.292 | 1.028 |
| Proposed (TV) | 1.053 | 0.261 | 1.349 | 0.243 |
| VB (C) | 0.032 | 0.022 | 0.015 | 0.037 |
| RS-IID (C) | 2.529 | 0.214 | 2.529 | 0.214 |
| Proposed (C) | 0.128 | 0.054 | 0.133 | 0.116 |

vious subsection. The bias values $\widehat{\mathbf{R}}^b(1,1)$ and $\widehat{\mathbf{R}}^b(1,2)$ for the filtered and smoothed MNC estimates are given in Table 4.7. This table reveals that the bias values of the VB method and the proposed method are similar and lower than the bias values of the RS-IID method for the time-varying MNC. In the constant MNC case, the VB method achieves the lowest bias values, whereas the RS-IID method has the highest bias values. This high bias in the covariance estimates results in low performance as seen in Tables 4.5 and 4.6. Furthermore, we again see that the time-varying covariance case has higher bias than the constant covariance case.

The results in this part show that the VB and the proposed solution provide similar results. The superiority of the proposed algorithm is the computation time. The computation times per MC run of the smoothers are given in Table 4.8.

## 4.8 Bayesian Filtering and Smoothing when both MNC and PNC are Unknown

When both noise covariances are unknown, it may be possible to obtain the filtering and smoothing solution by using the fixed point iteration (FPI) technique. We can first estimate the state and MNC. Then, we can estimate the state and PNC with found

Table 4.8: Computation times (per MC run) of the smoothers for the time-varying MNC scenario.

| Smoother | Computation Times (s) |
|---|---|
| RTS | $0.91$ |
| RS-IID | $29.02$ |
| VB | $27.85$ |
| Proposed | $7.43$ |

MNC. This iteration continues until convergence. At the end of iterations, we obtain the estimates for the state, PNC and MNC. However, the convergence of the mentioned FPI technique can be problematic. Although we just examine the smoothing problems with unknown PNC or unknown MNC, we have done several simulations with unknown PNC and MNC to check if the solutions converge or not. We observe from the simulations that the solutions converge. However, it is known that the FPI may fail to converge even when solving a simple equation like $q(x) = x$ where $q(x)$ is an arbitrary function and $x$ is just a scalar. Both the starting point of $x$ and the form of function $q(x)$ have an impact on convergence. Similar to this, we can encounter scenarios which may not produce convergent results. In that cases, we can set an iteration number at which we stop the iteration and provide the results. Another alternative can be to assign a covariance matrix to PNC or MNC, then find another covariance matrix with a single iteration. We should note that the performance of these two possible approaches will be worse than the case where both covariances are estimated with convergence. In cases where convergence is achieved, we can stop the iteration if the change in covariances is less than a certain threshold.

According to the discussions given above, we provide the pseudo-code of an algorithm solving the problem of Bayesian filtering and smoothing with unknown MNC and PNC. In order to show the convergence of the Algorithm 5, we use the same simulation setup as in Section 4.7 and we use time-varying MNC and PNC. In the simulations, we provide filtered and smoothed estimates for the first diagonal entries of the MNC and PNC. To demonstrate the convergence, the results are provided for

**Algorithm 5** Filtering and Smoothing with Unknown MNC and PNC

**Inputs:** $\mathbf{A}, \mathbf{C}, \mathbf{y}_{0:N}, J, \gamma_R, \gamma_Q, \hat{\mathbf{x}}^i, \mathbf{P}^i, v^i, \mathbf{V}^i, w^i, \mathbf{W}^i$.

Initialization: $\mathbf{V}_{n|N} = \mathbf{V}^i, v_{n|N} = v^i$ for $n = 0, 1, \ldots, N$, $w_{0|-1} = w^i, \mathbf{W}_{0|-1} = \mathbf{W}^i, v_{0|-1} = v^i, \mathbf{V}_{0|-1} = \mathbf{V}^i, \hat{\mathbf{x}}_{0|-1} = \hat{\mathbf{x}}^i, \mathbf{P}_{0|-1} = \mathbf{P}^i$.

**repeat**

  $\mathbf{Q}_n = \mathbf{V}_{n|N}/(v_{n|N} - 2n_x - 2)$ for $n = 0, 1, \ldots, N$.

  Run Algorithm 3 to obtain $\hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n}, w_{n|n}, \mathbf{W}_{n|n}$ for $n = 0, 1, \ldots, N$.

  Calculate $\hat{\mathbf{x}}_{n|N}, \mathbf{P}_{n|N}, w_{n|N}, \mathbf{W}_{n|N}$ for $n = N-1, N-2, \ldots, 0$ using (4.77).

  $\mathbf{R}_n = \mathbf{W}_{n|N}/(w_{n|N} - 2n_y - 2)$ for $n = 0, 1, \ldots, N$.

  Run Algorithm 2 to obtain $\hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n}, v_{n|n}, \mathbf{V}_{n|n}$ for $n = 0, 1, \ldots, N$.

  Run Algorithm 4 to obtain $\hat{\mathbf{x}}_{n|N}, \mathbf{P}_{n|N}, v_{n|N}, \mathbf{V}_{n|N}$ for $n = N-1, N-2, \ldots, 0$.

**until convergence**

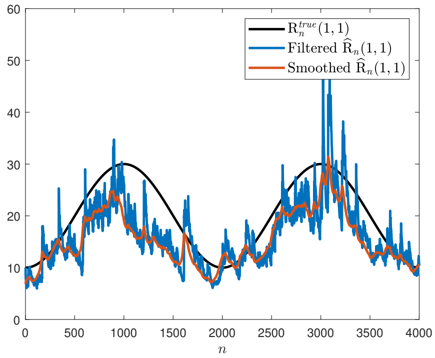
**Outputs:** $\hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n}, v_{n|n}, \mathbf{V}_{n|n}, w_{n|n}, \mathbf{W}_{n|n}, \hat{\mathbf{x}}_{n|N}, \mathbf{P}_{n|N}, v_{n|N}, \mathbf{V}_{n|N}, w_{n|N}, \mathbf{W}_{n|N}$ for $n = 0, 1, \ldots, N$.

the iteration numbers $1, 2, 4, 7, 9$ and $10$.

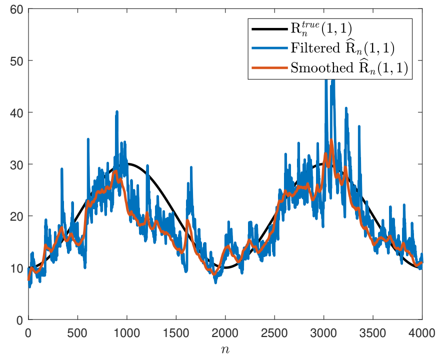
Figures 4.5 and 4.6 show that unknown MNC and PNC can be found iteratively and the convergence of the algorithm is fast.

### 4.9 Conclusions

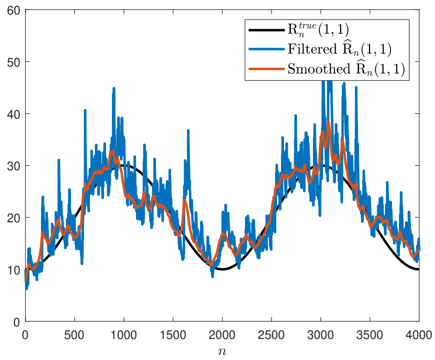
In this chapter, we have worked on filtering and smoothing problems. In the problem, we have used a linear state space representation and have assumed that the process and measurement noise covariances are unknown separately. The simulation results show the similar performance of the proposed methods with the existing alternatives. The methods are non-iterative unlike the existing approaches, which make them more efficient computationally. The proposed approaches are explicit in that they calculate explicit estimates of the unknown PNCs and MNCs, which might be useful in downstream applications in the data processing chain like clutter map formation (unknown MNC case) and/or target classification (unknown PNC case).



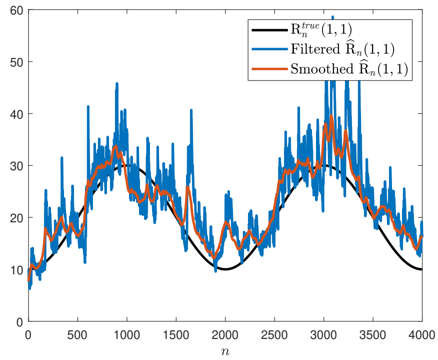

(a) $1^{st}$ iteration



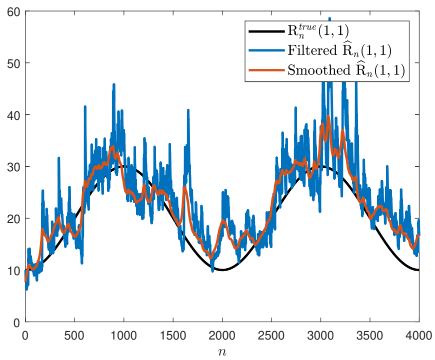

(b) $2^{nd}$ iteration



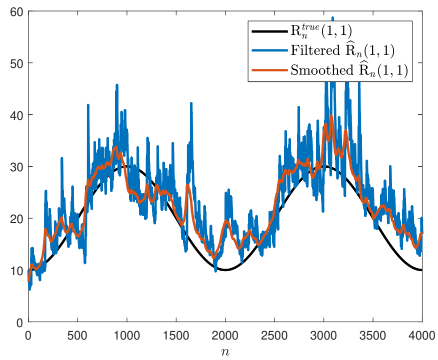

(c) $4^{th}$ iteration




(d) $7^{th}$ iteration

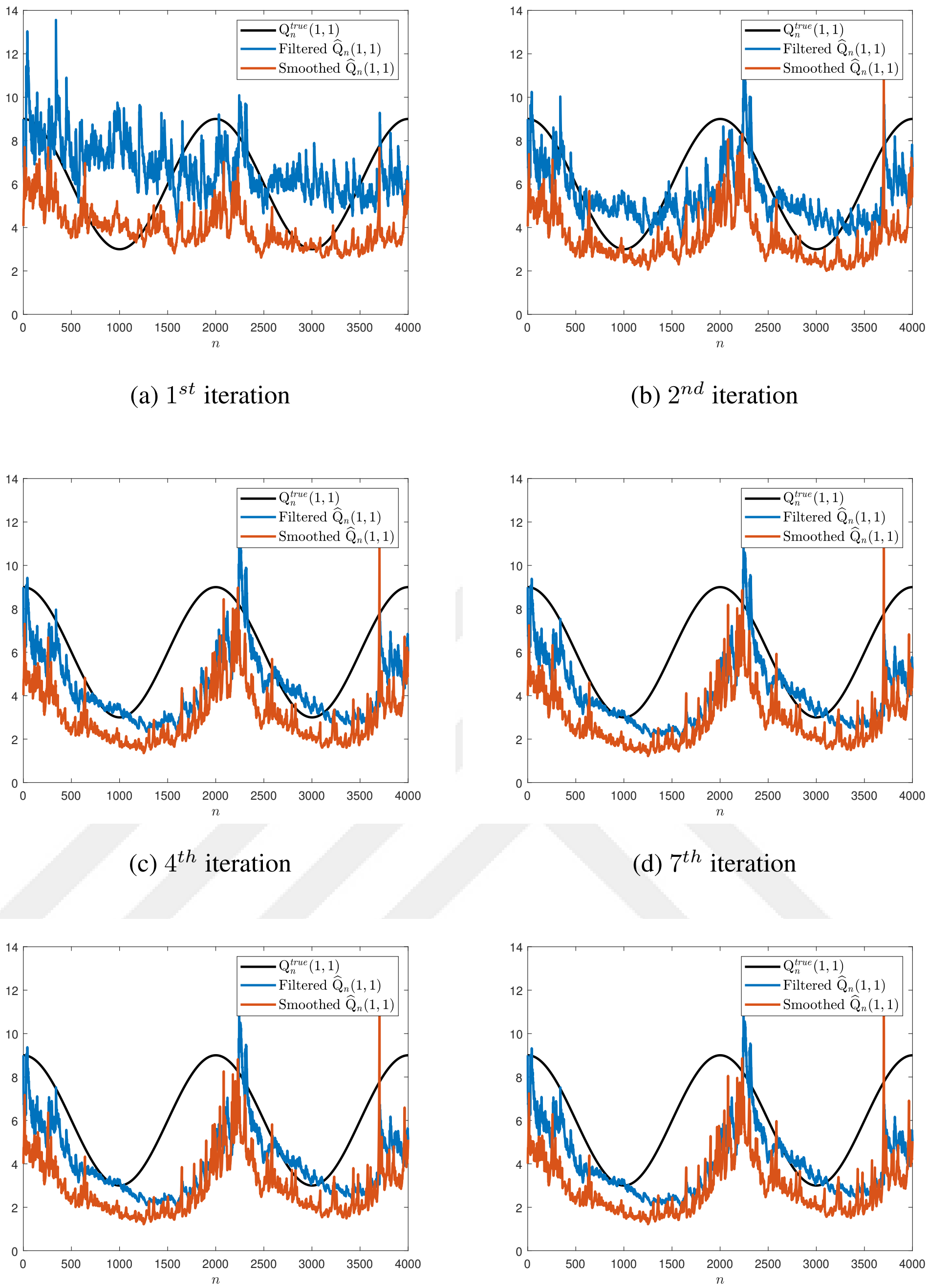



(e) $9^{th}$ iteration




(f) $10^{th}$ iteration

Figure 4.5: Filtered and smoothed estimates of $\widehat{\mathbf{R}}_n(1,1)$.




(a) $1^{st}$ iteration

(b) $2^{nd}$ iteration

(c) $4^{th}$ iteration

(d) $7^{th}$ iteration

(e) $9^{th}$ iteration

(f) $10^{th}$ iteration

Figure 4.6: Filtered and smoothed estimates of $\widehat{\mathbf{Q}}_n(1,1)$.

## 4.10 Derivations

### 4.10.1 Derivation of (4.9)

We can write the product $\mathcal{N}(\mathbf{x}_n; \mathbf{A}\mathbf{x}_{n-1}, \mathbf{Q}_n)\mathcal{IW}(\mathbf{Q}_n; v_{n|n-1}, \mathbf{V}_{n|n-1})$ as

$$\mathcal{N}(\mathbf{x}_n; \mathbf{A}\mathbf{x}_{n-1}, \mathbf{Q}_n)\mathcal{IW}(\mathbf{Q}_n; v_{n|n-1}, \mathbf{V}_{n|n-1})$$

$$\propto \frac{\mathcal{IW}(\mathbf{Q}_n; v_{n|n-1}+1, \mathbf{V}_{n|n-1} + (\mathbf{x}_n - \mathbf{A}\mathbf{x}_{n-1})(\cdot)^{\mathrm{T}})}{\left|\mathbf{V}_{n|n-1} + (\mathbf{x}_n - \mathbf{A}\mathbf{x}_{n-1})(\cdot)^{\mathrm{T}}\right|^{\frac{1}{2}(v_{n|n-1}-n_x)}} \tag{4.85a}$$

$$\propto \frac{\mathcal{IW}(\mathbf{Q}_n; v_{n|n-1}+1, \mathbf{V}_{n|n-1} + (\mathbf{x}_n - \mathbf{A}\mathbf{x}_{n-1})(\cdot)^{\mathrm{T}})}{\left|\mathcal{I}_{n_x} + \mathbf{V}_{n|n-1}^{-1/2}(\mathbf{x}_n - \mathbf{A}\mathbf{x}_{n-1})(\cdot)^{\mathrm{T}}\mathbf{V}_{n|n-1}^{-1/2}\right|^{\frac{1}{2}(v_{n|n-1}-n_x)}} \tag{4.85b}$$

$$\propto \frac{\mathcal{IW}(\mathbf{Q}_n; v_{n|n-1}+1, \mathbf{V}_{n|n-1} + (\mathbf{x}_n - \mathbf{A}\mathbf{x}_{n-1})(\cdot)^{\mathrm{T}})}{\left(1 + (\mathbf{x}_n - \mathbf{A}\mathbf{x}_{n-1})^{\mathrm{T}}\mathbf{V}_{n|n-1}^{-1}(\cdot)\right)^{\frac{1}{2}(v_{n|n-1}-n_x)}}, \tag{4.85c}$$

where we utilize the identity $|\mathcal{I} + \mathbf{A}\mathbf{B}| = |\mathcal{I} + \mathbf{B}\mathbf{A}|$ for any two matrices $\mathbf{A}$ and $\mathbf{B}$ for which the products $\mathbf{A}\mathbf{B}$ and $\mathbf{B}\mathbf{A}$ are well-defined. We use the expressions $(\mathbf{x})(\cdot)^{\mathrm{T}}$ and $(\mathbf{x})^{\mathrm{T}}\mathbf{A}(\cdot)$ for $(\mathbf{x})(\mathbf{x})^{\mathrm{T}}$ and $(\mathbf{x})^{\mathrm{T}}\mathbf{A}(\mathbf{x})$, respectively. We can represent the denominator of equation (4.85c) as multivariate Student's *t*-distribution. The general expression for this distribution is given as follows:

$$\mathrm{St}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu) = \frac{\left[1 + \frac{1}{\nu}(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right]^{-(\nu+p)/2}}{\left(\Gamma\left[(\nu+p)/2\right]\right)^{-1}\Gamma(\nu/2)\nu^{p/2}\pi^{p/2}|\boldsymbol{\Sigma}|^{1/2}} \tag{4.86}$$

where $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$, $\nu$ and $p$ represent location vector, scale matrix, degrees of freedom and the dimension of the vector $\mathbf{x}$, respectively. So, the final expression for the product can be written as

$$\begin{aligned}
\mathcal{N}(\mathbf{x}_n;&\mathbf{A}\mathbf{x}_{n-1}, \mathbf{Q}_n)\mathcal{IW}(\mathbf{Q}_n; v_{n|n-1}, \mathbf{V}_{n|n-1}) \\
&\propto \mathrm{St}\left(\mathbf{x}_n; \mathbf{A}\mathbf{x}_{n-1}, \frac{\mathbf{V}_{n|n-1}}{v_{n|n-1} - 2n_x}, v_{n|n-1} - 2n_x\right) \\
&\quad \times \mathcal{IW}(\mathbf{Q}_n; v_{n|n-1}+1, \mathbf{V}_{n|n-1} + (\mathbf{x}_n - \mathbf{A}\mathbf{x}_{n-1})(\mathbf{x}_n - \mathbf{A}\mathbf{x}_{n-1})^{\mathrm{T}}).
\end{aligned} \tag{4.87}$$

### 4.10.2 Moment Matching for an Infinite Mixture of IW Distributions

Let $\mathbf{X} \in \mathbb{R}^{d \times d}$ be distributed with the density $p(\mathbf{X})$ given as

$$p(\mathbf{X}) = \int r(\mathbf{x})\mathcal{IW}(\mathbf{X}; v, \mathbf{V}(\mathbf{x}))\,\mathrm{d}\mathbf{x}, \tag{4.88}$$

where $r(\cdot)$ is an arbitrary probability density function for the random vector $\mathbf{x}$ and $\mathbf{V}(\cdot)$ is an arbitrary positive definite function of $\mathbf{x}$. Suppose that we would like to approximate the density $p(\mathbf{X})$ with an IW distribution $q(\mathbf{X}) = \mathcal{IW}(\mathbf{X}; \overline{v}, \overline{\mathbf{V}})$ by using moment matching. The moments we would like to match are the mean $\mathrm{E}[\mathbf{X}]$ and the sum of the variances of the diagonal elements, i.e., $\sum_{m=1}^{d}\mathrm{Var}(\mathbf{X}_{mm})$. Notice that this selection is equivalent to matching the mean $\mathrm{E}[\mathbf{X}]$ and the expected sum of squared diagonal elements, i.e., $\mathrm{E}[\sum_{m=1}^{d}\mathbf{X}_{mm}^2]$.

We can calculate the means of the distributions $p(\mathbf{X})$ and $q(\mathbf{X})$ as

$$\mathrm{E}_p(\mathbf{X}) = \frac{\int r(\mathbf{x})\mathbf{V}(\mathbf{x})\,\mathrm{d}\mathbf{x}}{v-2d-2} = \frac{\mathrm{E}_r[\mathbf{V}(\mathbf{x})]}{u}, \tag{4.89a}$$

$$\mathrm{E}_q(\mathbf{X}) = \frac{\overline{\mathbf{V}}}{\overline{v}-2d-2} = \frac{\overline{\mathbf{V}}}{\overline{u}}, \tag{4.89b}$$

where $u = v-2d-2$ and $\overline{u} = \overline{v}-2d-2$. Solving the equation $\mathrm{E}_q(\mathbf{X}) = \mathrm{E}_p(\mathbf{X})$ for $\overline{\mathbf{V}}$ gives

$$\overline{\mathbf{V}} = \frac{\overline{v}-2d-2}{v-2d-2}\,\mathrm{E}_r\left[\mathbf{V}(\mathbf{x})\right]. \tag{4.90}$$

In [6, Theorem 3.4.3], the variance of the $m$th diagonal element, i.e., $\mathrm{Var}(\mathbf{X}_{mm})$, of $\mathbf{X} \sim \mathcal{IW}(\mathbf{X}; v, \mathbf{V})$ is given as

$$\mathrm{Var}(\mathbf{X}_{mm}) = \frac{\left(\frac{2}{v-2d-2}+2\right)\mathbf{V}_{mm}^2}{(v-2d-1)(v-2d-2)(v-2d-4)} \tag{4.91a}$$

$$= \frac{\left(\frac{2}{u}+2\right)\mathbf{V}_{mm}^2}{(u+1)u(u-2)} = \frac{2\mathbf{V}_{mm}^2}{u^2(u-2)}, \tag{4.91b}$$

where $u = v-2d-2$. Based on this, we can find $\mathrm{E}(\mathbf{X}_{mm}^2)$ as follows.

$$\mathrm{E}(\mathbf{X}_{mm}^2) = \frac{2\mathbf{V}_{mm}^2}{u^2(u-2)} + \frac{\mathbf{V}_{mm}^2}{u^2} = \frac{\mathbf{V}_{mm}^2}{u(u-2)}, \tag{4.92}$$

which is written by using the identity $\mathrm{Var}(\mathbf{X}_{mm}) = \mathrm{E}(\mathbf{X}_{mm}^2) - \mathrm{E}^2(\mathbf{X}_{mm})$. Hence, the expectation of sum of squared diagonal elements can be given as

$$\mathrm{E}\left[\sum_{m=1}^{d}\mathbf{X}_{mm}^2\right] = \frac{1}{u(u-2)}\sum_{m=1}^{d}\mathbf{V}_{mm}^2. \tag{4.93}$$

We can now write the expectations of the sum of squared diagonal elements with respect to $p(\mathbf{X})$ and $q(\mathbf{X})$ as follows.

$$\mathrm{E}_p\left[\sum_{m=1}^{d}\mathbf{X}_{mm}^2\right] = \frac{1}{u(u-2)}\sum_{m=1}^{d}\mathrm{E}_r\left[\mathbf{V}_{mm}^2(\mathbf{x})\right], \tag{4.94a}$$

$$\mathrm{E}_q\left[\sum_{m=1}^{d}\mathbf{X}_{mm}^2\right] = \frac{1}{\overline{u}(\overline{u}-2)}\sum_{m=1}^{d}\overline{\mathbf{V}}_{mm}^2. \tag{4.94b}$$

Equating (4.94a) to (4.94b) and solving for $\overline{v}$ gives

$$\overline{v} = 2d+2+\frac{2(v-2d-2)}{2\alpha+(1-\alpha)(v-2d-2)}, \tag{4.95}$$

where

$$\alpha = \frac{\sum_{m=1}^{d}\mathrm{E}_r^2\left[\mathbf{V}_{mm}(\mathbf{x})\right]}{\sum_{m=1}^{d}\mathrm{E}_r\left[\mathbf{V}_{mm}^2(\mathbf{x})\right]}. \tag{4.96}$$

Note that due to Jensen's inequality [92], we have $\alpha \leq 1$.

### 4.10.3 Approximation of Joint Predicted Distribution

In this section, we find an approximation for the joint predicted density $p(\mathbf{x}_{n+1}, \mathbf{Q}_{n+1}|\mathbf{y}_{0:n})$. We write $p(\mathbf{x}_{n+1}, \mathbf{Q}_{n+1}|\mathbf{y}_{0:n})$ by using the Chapman-Kolmogorov equation as follows:

$$p(\mathbf{x}_{n+1}, \mathbf{Q}_{n+1}|\mathbf{y}_{0:n}) = \int\int p(\mathbf{x}_{n+1}, \mathbf{Q}_{n+1}|\mathbf{x}_n, \mathbf{Q}_n)p(\mathbf{x}_n, \mathbf{Q}_n|\mathbf{y}_{0:n})\, \mathrm{d}\mathbf{x}_n\, \mathrm{d}\mathbf{Q}_n. \tag{4.97}$$

The terms on the right hand side of (4.97) can be written as

$$p(\mathbf{x}_{n+1}, \mathbf{Q}_{n+1}|\mathbf{x}_n, \mathbf{Q}_n) = p(\mathbf{x}_{n+1}|\mathbf{x}_n, \mathbf{Q}_{n+1})p(\mathbf{Q}_{n+1}|\mathbf{Q}_n), \tag{4.98a}$$

$$p(\mathbf{x}_n, \mathbf{Q}_n|\mathbf{y}_{0:n}) \approx p(\mathbf{x}_n|\mathbf{y}_{0:n})p(\mathbf{Q}_n|\mathbf{y}_{0:n}). \tag{4.98b}$$

$p(\mathbf{x}_{n+1}, \mathbf{Q}_{n+1}|\mathbf{y}_{0:n})$ can be rewritten by substituting the expressions in (4.98) into (4.97) as follows:

$$p(\mathbf{x}_{n+1}, \mathbf{Q}_{n+1}|\mathbf{y}_{0:n}) = \int\int p(\mathbf{x}_{n+1}|\mathbf{x}_n, \mathbf{Q}_{n+1})p(\mathbf{Q}_{n+1}|\mathbf{Q}_n) \times p(\mathbf{x}_n|\mathbf{y}_{0:n})p(\mathbf{Q}_n|\mathbf{y}_{0:n})\, \mathrm{d}\mathbf{x}_n\, \mathrm{d}\mathbf{Q}_n. \tag{4.99}$$

The equation (4.99) shows that analytical evaluation of $p(\mathbf{x}_{n+1}, \mathbf{Q}_{n+1}|\mathbf{y}_{0:n})$ is not possible. Therefore, we perform the following approximation.

$$p(\mathbf{x}_{n+1}, \mathbf{Q}_{n+1}|\mathbf{y}_{0:n}). \approx p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n})p(\mathbf{Q}_{n+1}|\mathbf{y}_{0:n}). \tag{4.100}$$

The terms $p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n})$ and $p(\mathbf{Q}_{n+1}|\mathbf{y}_{0:n})$ can be found by marginalizing the joint predicted density $p(\mathbf{x}_{n+1}, \mathbf{Q}_{n+1}|\mathbf{y}_{0:n})$. We first find the density $p(\mathbf{Q}_{n+1}|\mathbf{y}_{0:n})$ as

$$p(\mathbf{Q}_{n+1}|\mathbf{y}_{0:n}) = \int p(\mathbf{x}_{n+1}, \mathbf{Q}_{n+1}|\mathbf{y}_{0:n})\, \mathrm{d}\mathbf{x}_{n+1} \tag{4.101a}$$

$$= \int\int\int p(\mathbf{x}_{n+1}|\mathbf{x}_n, \mathbf{Q}_{n+1})p(\mathbf{Q}_{n+1}|\mathbf{Q}_n)p(\mathbf{x}_n|\mathbf{y}_{0:n})p(\mathbf{Q}_n|\mathbf{y}_{0:n})\, \mathrm{d}\mathbf{x}_n\, \mathrm{d}\mathbf{Q}_n\, \mathrm{d}\mathbf{x}_{n+1} \tag{4.101b}$$

$$= \int\int p(\mathbf{Q}_{n+1}|\mathbf{Q}_n)p(\mathbf{x}_n|\mathbf{y}_{0:n})p(\mathbf{Q}_n|\mathbf{y}_{0:n})\, \mathrm{d}\mathbf{x}_n\, \mathrm{d}\mathbf{Q}_n \tag{4.101c}$$

$$= \int p(\mathbf{Q}_{n+1}|\mathbf{Q}_n)p(\mathbf{Q}_n|\mathbf{y}_{0:n})\, \mathrm{d}\mathbf{Q}_n \tag{4.101d}$$

$$= \int p(\mathbf{Q}_{n+1}|\mathbf{Q}_n)\mathcal{IW}(\mathbf{Q}_n; v_{n|n}, \mathbf{V}_{n|n})\, \mathrm{d}\mathbf{Q}_n \tag{4.101e}$$

$$=\mathcal{IW}(\mathbf{Q}_{n+1}; v_{n+1|n}, \mathbf{V}_{n+1|n}), \tag{4.101f}$$

where

$$v_{n+1|n} \triangleq \gamma_Q v_{n|n} + (1-\gamma_Q)(2n_x+2), \qquad \mathbf{V}_{n+1|n} \triangleq \gamma_Q \mathbf{V}_{n|n}. \tag{4.102}$$

The predicted state density $p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n})$ can be written as

$$p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n}) = \int p(\mathbf{x}_{n+1}, \mathbf{Q}_{n+1}|\mathbf{y}_{0:n})\, \mathrm{d}\mathbf{Q}_{n+1} \tag{4.103a}$$

$$= \int\int\int p(\mathbf{x}_{n+1}|\mathbf{x}_n, \mathbf{Q}_{n+1}) p(\mathbf{Q}_{n+1}|\mathbf{Q}_n) p(\mathbf{x}_n|\mathbf{y}_{0:n}) p(\mathbf{Q}_n|\mathbf{y}_{0:n})\, \mathrm{d}\mathbf{x}_n\, \mathrm{d}\mathbf{Q}_n\, \mathrm{d}\mathbf{Q}_{n+1} \tag{4.103b}$$

$$= \int\int\int \mathcal{N}(\mathbf{x}_{n+1}; \mathbf{A}\mathbf{x}_n, \mathbf{Q}_{n+1}) p(\mathbf{Q}_{n+1}|\mathbf{Q}_n) \times \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n}) \mathcal{IW}(\mathbf{Q}_n; v_{n|n}, \mathbf{V}_{n|n})\, \mathrm{d}\mathbf{x}_n\, \mathrm{d}\mathbf{Q}_n\, \mathrm{d}\mathbf{Q}_{n+1} \tag{4.103c}$$

$$= \int\int \mathcal{N}(\mathbf{x}_{n+1}; \mathbf{A}\mathbf{x}_n, \mathbf{Q}_{n+1}) \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n}) \times \mathcal{IW}(\mathbf{Q}_{n+1}; v_{n+1|n}, \mathbf{V}_{n+1|n})\, \mathrm{d}\mathbf{x}_n\, \mathrm{d}\mathbf{Q}_{n+1} \tag{4.103d}$$

$$\overset{\times}{=} \int\int \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n}) \times \frac{\mathcal{IW}\left(\mathbf{Q}_{n+1}; v_{n+1|n}+1, \mathbf{V}_{n+1|n} + (\mathbf{x}_{n+1} - \mathbf{A}\mathbf{x}_n)(\cdot)^{\mathrm{T}}\right)}{\left(1 + (\mathbf{x}_{n+1} - \mathbf{A}\mathbf{x}_n)^{\mathrm{T}} \mathbf{V}_{n+1|n}^{-1}(\cdot)\right)^{\frac{1}{2}(v_{n+1|n} - n_x)}}\, \mathrm{d}\mathbf{x}_n\, \mathrm{d}\mathbf{Q}_{n+1} \tag{4.103e}$$

$$= \int \frac{\mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n})}{\left(1 + (\mathbf{x}_{n+1} - \mathbf{A}\mathbf{x}_n)^{\mathrm{T}} \mathbf{V}_{n+1|n}^{-1}(\cdot)\right)^{\frac{1}{2}(v_{n+1|n} - n_x)}}\, \mathrm{d}\mathbf{x}_n \tag{4.103f}$$

$$\overset{\times}{=} \int \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n}) \mathrm{St}\left(\mathbf{x}_{n+1}; \mathbf{A}\mathbf{x}_n, \frac{\mathbf{V}_{n+1|n}}{v_{n+1|n} - 2n_x}, v_{n+1|n} - 2n_x\right)\, \mathrm{d}\mathbf{x}_n. \tag{4.103g}$$

We can now approximate $p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n})$ using moment matching as $p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n}) \approx \mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|n}, \hat{\mathbf{P}}_{n+1|n})$ where $\hat{\mathbf{x}}_{n+1|n}$ and $\hat{\mathbf{P}}_{n+1|n}$ can be calculated as follows:

$$\hat{\mathbf{x}}_{n+1|n} = \int \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n}) \int \mathbf{x}_{n+1} \times \mathrm{St}\left(\mathbf{x}_{n+1}; \mathbf{A}\mathbf{x}_n, \frac{\mathbf{V}_{n+1|n}}{v_{n+1|n} - 2n_x}, v_{n+1|n} - 2n_x\right)\, \mathrm{d}\mathbf{x}_{n+1}\, \mathrm{d}\mathbf{x}_n \tag{4.104a}$$

$$= \int \mathbf{A}\mathbf{x}_n \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n})\, \mathrm{d}\mathbf{x}_n \tag{4.104b}$$

$$= \mathbf{A}\hat{\mathbf{x}}_{n|n}, \tag{4.104c}$$

$$\hat{\mathbf{P}}_{n+1|n} = \int\int (\mathbf{x}_{n+1} - \mathbf{A}\hat{\mathbf{x}}_{n|n})(\mathbf{x}_{n+1} - \mathbf{A}\hat{\mathbf{x}}_{n|n})^{\mathrm{T}} \times \mathrm{St}\left(\mathbf{x}_{n+1}; \mathbf{A}\mathbf{x}_n, \frac{\mathbf{V}_{n+1|n}}{v_{n+1|n} - 2n_x}, v_{n+1|n} - 2n_x\right)\, \mathrm{d}\mathbf{x}_{n+1}$$

$$\times \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n})\,\mathrm{d}\mathbf{x}_n \tag{4.104d}$$

$$= \int \left( \frac{\mathbf{V}_{n+1|n}}{\upsilon_{n+1|n} - 2n_x - 2} + \mathbf{A}(\mathbf{x}_n - \hat{\mathbf{x}}_{n|n})(\cdot)^{\mathrm{T}}\mathbf{A}^{\mathrm{T}} \right) \mathcal{N}(\mathbf{x}_n; \hat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n})\,\mathrm{d}\mathbf{x}_n \tag{4.104e}$$

$$= \mathbf{A}\mathbf{P}_{n|n}\mathbf{A}^{\mathrm{T}} + \hat{\mathbf{Q}}_{n+1|n}, \tag{4.104f}$$

where

$$\hat{\mathbf{Q}}_{n+1|n} = \frac{\mathbf{V}_{n+1|n}}{\upsilon_{n+1|n} - 2n_x - 2}. \tag{4.105}$$

As a result, the predicted joint density $p(\mathbf{x}_{n+1}, \mathbf{Q}_{n+1}|\mathbf{y}_{0:n})$ is given as

$$\begin{aligned} p(\mathbf{x}_{n+1}, \mathbf{Q}_{n+1}|\mathbf{y}_{0:n}) \approx & \mathcal{N}(\mathbf{x}_{n+1}; \hat{\mathbf{x}}_{n+1|n}, \mathbf{A}\mathbf{P}_{n|n}\mathbf{A}^{\mathrm{T}} + \hat{\mathbf{Q}}_{n+1|n}) \\ & \times \mathcal{IW}(\mathbf{Q}_{n+1}; \upsilon_{n+1|n}, \mathbf{V}_{n+1|n}). \end{aligned} \tag{4.106}$$

### 4.10.4 Calculation of $\mathrm{E}[\mathbf{V}(\mathbf{w}_n)]$ and $\mathrm{E}[\mathbf{V}^2_{mm}(\mathbf{w}_n)]$ for Smoothing with Unknown PNC

In this part, we derive approximate analytical expressions for $\mathrm{E}[\mathbf{V}(\mathbf{w}_n)]$ and $\mathrm{E}[\mathbf{V}^2_{mm}(\mathbf{w}_n)]$. We first use the matrix inversion lemma two times for $\mathbf{V}(\mathbf{w}_n)$.

$$\mathbf{V}(\mathbf{w}_n) = \left[(1-\gamma_Q)\mathbf{V}^{-1}_{n|n} + \gamma_Q(\mathbf{V}_{n+1|N} + \mathbf{w}_n\mathbf{w}_n^{\mathrm{T}})^{-1}\right]^{-1} \tag{4.107a}$$

$$= \left[(1-\gamma_Q)\mathbf{V}^{-1}_{n|n} + \gamma_Q\left(\mathbf{V}^{-1}_{n+1|N} - \frac{\mathbf{V}^{-1}_{n+1|N}\mathbf{w}_n\mathbf{w}_n^{\mathrm{T}}\mathbf{V}^{-1}_{n+1|N}}{1 + \mathbf{w}_n^{\mathrm{T}}\mathbf{V}^{-1}_{n+1|N}\mathbf{w}_n}\right)\right]^{-1} \tag{4.107b}$$

$$= \left[(1-\gamma_Q)\mathbf{V}^{-1}_{n|n} + \gamma_Q\mathbf{V}^{-1}_{n+1|N} - \frac{\gamma_Q\mathbf{V}^{-1}_{n+1|N}\mathbf{w}_n\mathbf{w}_n^{\mathrm{T}}\mathbf{V}^{-1}_{n+1|N}}{1 + \mathbf{w}_n^{\mathrm{T}}\mathbf{V}^{-1}_{n+1|N}\mathbf{w}_n}\right]^{-1} \tag{4.107c}$$

$$= \left[\overline{\mathbf{V}}^{-1} - \frac{\gamma_Q}{1 + \mathbf{w}_n^{\mathrm{T}}\mathbf{V}^{-1}_{n+1|N}\mathbf{w}_n}\mathbf{V}^{-1}_{n+1|N}\mathbf{w}_n\mathbf{w}_n^{\mathrm{T}}\mathbf{V}^{-1}_{n+1|N}\right]^{-1} \tag{4.107d}$$

$$= \overline{\mathbf{V}} + \frac{\overline{\mathbf{V}}\mathbf{V}^{-1}_{n+1|N}\mathbf{w}_n\mathbf{w}_n^{\mathrm{T}}\mathbf{V}^{-1}_{n+1|N}\overline{\mathbf{V}}}{\gamma_Q^{-1}(1 + \mathbf{w}_n^{\mathrm{T}}\mathbf{V}^{-1}_{n+1|N}\mathbf{w}_n) - \mathbf{w}_n^{\mathrm{T}}\mathbf{V}^{-1}_{n+1|N}\overline{\mathbf{V}}\mathbf{V}^{-1}_{n+1|N}\mathbf{w}_n} \tag{4.107e}$$

$$= \overline{\mathbf{V}} + \gamma_Q\frac{\overline{\mathbf{V}}\mathbf{V}^{-1}_{n+1|N}\mathbf{w}_n\mathbf{w}_n^{\mathrm{T}}\mathbf{V}^{-1}_{n+1|N}\overline{\mathbf{V}}}{1 + \mathbf{w}_n^{\mathrm{T}}\mathbf{V}^{-1}_{n+1|N}(\mathbf{V}_{n+1|N} - \gamma_Q\overline{\mathbf{V}})\mathbf{V}^{-1}_{n+1|N}\mathbf{w}_n}, \tag{4.107f}$$

where $\overline{\mathbf{V}}$ is defined as

$$\overline{\mathbf{V}} = \left((1-\gamma_Q)\mathbf{V}^{-1}_{n|n} + \gamma_Q\mathbf{V}^{-1}_{n+1|N}\right)^{-1}. \tag{4.108}$$

We would like to find the expected value of $\mathbf{V}(\mathbf{w}_n)$ with respect to $\mathbf{w}_n$ which has a Gaussian mixture distribution $p(\mathbf{w}_n)$ given in (4.75). For the sake of brevity, we take expectation with respect to a Gaussian distribution as it is easy to find the final expected value with a weighted sum of the expected values of $\mathbf{V}(\mathbf{w}_n)$ with respect to components of $p(\mathbf{w}_n)$. Hence, we take expectation with respect to $\mathbf{w}_n \sim \mathcal{N}(\mathbf{w}_n; \overline{\mathbf{w}}, \overline{\mathbf{W}})$. To get a simpler form for (4.107f), we define $\mathbf{u} \triangleq \overline{\mathbf{V}}\mathbf{V}_{n+1|N}^{-1}\mathbf{w}_n$. Then, we write $\mathbf{V}(\mathbf{u})$ as

$$\mathbf{V}(\mathbf{u}) = \overline{\mathbf{V}} + \gamma_Q \frac{\mathbf{u}\mathbf{u}^{\mathrm{T}}}{1 + \mathbf{u}^{\mathrm{T}}\overline{\mathbf{V}}^{-1}(\mathbf{V}_{n+1|N} - \gamma_Q\overline{\mathbf{V}})\overline{\mathbf{V}}^{-1}\mathbf{u}} \tag{4.109}$$

Now, we define $\mathbf{U} \triangleq \overline{\mathbf{V}}^{-1}(\mathbf{V}_{n+1|N} - \gamma_Q\overline{\mathbf{V}})\overline{\mathbf{V}}^{-1}$, and $\mathbf{V}(\mathbf{u})$ can be expressed as

$$\mathbf{V}(\mathbf{u}) = \overline{\mathbf{V}} + \gamma_Q \frac{\mathbf{u}\mathbf{u}^{\mathrm{T}}}{1 + \mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}}, \tag{4.110}$$

where $\mathbf{u}$ has a Gaussian distribution

$$\mathbf{u} \sim \mathcal{N}(\mathbf{u}; \overline{\mathbf{u}}, \overline{\mathbf{U}}), \tag{4.111}$$

where

$$\overline{\mathbf{u}} \triangleq \overline{\mathbf{V}}\mathbf{V}_{n+1|N}^{-1}\overline{\mathbf{w}}, \tag{4.112a}$$

$$\overline{\mathbf{U}} \triangleq \overline{\mathbf{V}}\mathbf{V}_{n+1|N}^{-1}\overline{\mathbf{W}}\mathbf{V}_{n+1|N}^{-1}\overline{\mathbf{V}}. \tag{4.112b}$$

We first start with the calculation of $\mathrm{E}[\mathbf{V}(\mathbf{u})]$.

$$\mathrm{E}[\mathbf{V}(\mathbf{u})] = \overline{\mathbf{V}} + \gamma_Q \,\mathrm{E}\left[\frac{\mathbf{u}\mathbf{u}^{\mathrm{T}}}{1 + \mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}}\right]. \tag{4.113}$$

In order to calculate the expectation, we use the following theorem.

**Theorem 1.**

$$\mathrm{E}\left[\frac{\mathbf{u}\mathbf{u}^{\mathrm{T}}}{1 + \mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}}\right] = \frac{\partial}{\partial \mathbf{U}}\,\mathrm{E}[\log(1 + \mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u})]. \tag{4.114}$$

Hence, if we can find $\mathrm{E}[\log(1 + \mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u})]$, we get the desired expected value by taking the derivative of the result with respect to matrix $\mathbf{U}$. Since we cannot calculate $\mathrm{E}[\log(1 + \mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u})]$ analytically, we approximate the term inside the expectation operation as follows:

$$\log(1 + x) \approx \log(1 + \mathrm{E}[x]) + \frac{1}{1 + \mathrm{E}[x]}(x - \mathrm{E}[x])$$

$$- \frac{1}{2(1+\mathrm{E}[x])^2}(x - \mathrm{E}[x])^2. \tag{4.115}$$

The expectation of this expression is provided as

$$\mathrm{E}[\log(1+x)] \approx \log(1+\mathrm{E}[x]) - \frac{\mathrm{Var}(x)}{2(1+\mathrm{E}[x])^2}. \tag{4.116}$$

By using this approximation, we can write $\mathrm{E}[\log(1+\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u})]$ as

$$\mathrm{E}[\log(1+\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u})] \approx \log(1+\mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}]) - \frac{\mathrm{Var}(\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u})}{2(1+\mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}])^2}. \tag{4.117}$$

The expectation and variance expressions given in the above equation can be calculated as follows:

$$\begin{aligned} \mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}] &= \bar{\mathbf{u}}^{\mathrm{T}}\mathbf{U}\bar{\mathbf{u}} + \mathrm{tr}(\mathbf{U}\overline{\mathbf{U}}), \\ \mathrm{Var}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}] &= 2\,\mathrm{tr}(\mathbf{U}\overline{\mathbf{U}}\mathbf{U}\overline{\mathbf{U}}) + 4\bar{\mathbf{u}}^{\mathrm{T}}\mathbf{U}\overline{\mathbf{U}}\mathbf{U}\bar{\mathbf{u}}. \end{aligned} \tag{4.118}$$

We can now write the derivative of $\mathrm{E}[\log(1+\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u})]$ with respect to $\mathbf{U}$ as follows:

$$\begin{aligned} \frac{\partial}{\partial \mathbf{U}} \mathrm{E}[\log(1+\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u})] =& \frac{\frac{\partial}{\partial \mathbf{U}} \mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}]}{1+\mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}]} - \frac{1}{2}\frac{\frac{\partial}{\partial \mathbf{U}} \mathrm{Var}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}]}{(1+\mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}])^2} \\ &+ \frac{\mathrm{Var}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}]\frac{\partial}{\partial \mathbf{U}} \mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}]}{(1+\mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}])^3}, \end{aligned} \tag{4.119}$$

where

$$\begin{aligned} \frac{\partial}{\partial \mathbf{U}} \mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}] &= \overline{\mathbf{U}} + \bar{\mathbf{u}}\bar{\mathbf{u}}^{\mathrm{T}}, \\ \frac{\partial}{\partial \mathbf{U}} \mathrm{Var}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}] &= 4\overline{\mathbf{U}}\mathbf{U}\overline{\mathbf{U}} + 8\overline{\mathbf{U}}\mathbf{U}\bar{\mathbf{u}}\bar{\mathbf{u}}^{\mathrm{T}}. \end{aligned} \tag{4.120}$$

The final expression for $\mathrm{E}(\mathbf{V}(\mathbf{u}))$ can be expressed as

$$\begin{aligned} \mathrm{E}[\mathbf{V}(\mathbf{u})] = \overline{\mathbf{V}} + \gamma_Q \Bigg[ & \frac{\overline{\mathbf{U}} + \bar{\mathbf{u}}\bar{\mathbf{u}}^{\mathrm{T}}}{1+\mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}]} - \frac{1}{2}\frac{4\overline{\mathbf{U}}\mathbf{U}\overline{\mathbf{U}} + 8\overline{\mathbf{U}}\mathbf{U}\bar{\mathbf{u}}\bar{\mathbf{u}}^{\mathrm{T}}}{(1+\mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}])^2} \\ &+ \frac{\mathrm{Var}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}](\overline{\mathbf{U}} + \bar{\mathbf{u}}\bar{\mathbf{u}}^{\mathrm{T}})}{(1+\mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}])^3} \Bigg]. \end{aligned} \tag{4.121}$$

It should be clear that

$$\mathbf{V}_{mm}(\mathbf{u}) = \overline{\mathbf{V}}_{mm} + \gamma_Q \frac{\mathbf{u}_m^2}{1+\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}}. \tag{4.122}$$

We take the expectation of this expression as

$$\mathrm{E}[\mathbf{V}_{mm}(\mathbf{u})] = \overline{\mathbf{V}}_{mm} + \gamma_Q \left[ \mathrm{E}\left[ \frac{\mathbf{u}\mathbf{u}^{\mathrm{T}}}{1+\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}} \right] \right]_{mm} \tag{4.123a}$$

$$
\begin{aligned}
=&\overline{\mathbf{V}}_{mm} + \gamma_Q \Bigg[ \frac{\overline{\mathbf{U}}_{mm} + \overline{\mathbf{u}}_m^2}{1 + \mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}]} - \frac{2\overline{\mathbf{U}}_{:,m}^{\mathrm{T}}\mathbf{U}\overline{\mathbf{U}}_{:,m} + 4\overline{\mathbf{U}}_{:,m}^{\mathrm{T}}\mathbf{U}\bar{\mathbf{u}}\bar{\mathbf{u}}_m}{(1 + \mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}])^2} \\
&+ \frac{\mathrm{Var}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}](\overline{\mathbf{U}}_{mm} + \overline{\mathbf{u}}_m^2)}{(1 + \mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}])^3} \Bigg],
\end{aligned} \tag{4.123b}
$$

where $\overline{\mathbf{U}}_{:,m}$ is the $m^{th}$ column of matrix $\overline{\mathbf{U}}$.

The last identity that we need to find is $\mathrm{E}[\mathbf{V}_{mm}^2(\mathbf{u})]$.

$$
\mathrm{E}[\mathbf{V}_{mm}^2(\mathbf{u})] = \overline{\mathbf{V}}_{mm}^2 + 2\gamma_Q \overline{\mathbf{V}}_{mm} \, \mathrm{E}\left[ \frac{\mathbf{u}_m^2}{1 + \mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}} \right] + \gamma_Q^2 \, \mathrm{E}\left[ \frac{\mathbf{u}_m^4}{(1 + \mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u})^2} \right]. \tag{4.124}
$$

Since we have found the second term given on the right-hand side of (4.124), the only expression that we need to calculate is the third term. For this purpose, we use the following theorem.

**Theorem 2.**

$$
\mathrm{E}\left[ \frac{\mathbf{u}_m^4}{(1 + \mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u})^2} \right] = -\frac{\partial}{\partial \mathbf{U}_{mm}} \mathrm{E}\left[ \frac{\mathbf{u}_m^2}{1 + \mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}} \right]. \tag{4.125}
$$

We calculate the right hand side in (4.125) as follows:

$$
\begin{aligned}
\frac{\partial}{\partial \mathbf{U}_{mm}} \mathrm{E}\left[ \frac{\mathbf{u}_m^2}{1 + \mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}} \right] = \frac{\partial}{\partial \mathbf{U}_{mm}} \Bigg[ & \frac{\left[ \frac{\partial}{\partial \mathbf{U}} \mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}] \right]_{mm}}{1 + \mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}]} \\
& - \frac{1}{2} \frac{\left[ \frac{\partial}{\partial \mathbf{U}} \mathrm{Var}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}] \right]_{mm}}{(1 + \mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}])^2} + \frac{\mathrm{Var}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}] \left[ \frac{\partial}{\partial \mathbf{U}} \mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}] \right]_{mm}}{(1 + \mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}])^3} \Bigg]
\end{aligned} \tag{4.126a}
$$

$$
\begin{aligned}
= & - \frac{\left( \frac{\partial}{\partial \mathbf{U}_{mm}} \mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}] \right)^2}{(1 + \mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}])^2} - \frac{1}{2} \frac{\frac{\partial^2}{\partial \mathbf{U}_{mm}^2} \mathrm{Var}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}]}{(1 + \mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}])^2} \\
& + 2 \frac{\frac{\partial}{\partial \mathbf{U}_{mm}} \mathrm{Var}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}] \frac{\partial}{\partial \mathbf{U}_{mm}} \mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}]}{(1 + \mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}])^3} \\
& - 3 \frac{\mathrm{Var}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}] \left( \frac{\partial}{\partial \mathbf{U}_{mm}} \mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}] \right)^2}{(1 + \mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}])^4},
\end{aligned} \tag{4.126b}
$$

where

$$
\frac{\partial}{\partial \mathbf{U}_{mm}} \mathrm{E}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}] = \overline{\mathbf{U}}_{mm} + \overline{\mathbf{u}}_m^2, \tag{4.127a}
$$

$$
\frac{\partial}{\partial \mathbf{U}_{mm}} \mathrm{Var}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}] = 4\overline{\mathbf{U}}_{:,m}^{\mathrm{T}}\mathbf{U}\overline{\mathbf{U}}_{:,m} + 8\overline{\mathbf{U}}_{:,m}^{\mathrm{T}}\mathbf{U}\bar{\mathbf{u}}\bar{\mathbf{u}}_m, \tag{4.127b}
$$

$$
\frac{\partial^2}{\partial \mathbf{U}_{mm}^2} \mathrm{Var}[\mathbf{u}^{\mathrm{T}}\mathbf{U}\mathbf{u}] = 4\overline{\mathbf{U}}_{mm}^2 + 8\overline{\mathbf{U}}_{mm}\overline{\mathbf{u}}_m^2. \tag{4.127c}
$$

# CHAPTER 5

# CONCLUSIONS AND FUTURE WORK

In this thesis, we have worked on the problem of Bayesian state estimation. We have looked for an answer to the question of whether it is possible to develop a measurement update rule that performs almost as well as a particle filter without using Monte Carlo techniques. Another issue we have looked at was whether recursive filtering/smoothing could be achieved while simultaneously attempting to estimate measurement or process noise covariance in the scenario where one of these covariances is unknown.

We have first examined the filtering problem with nonlinear measurements. We have solved the problem by minimizing the Kullback-Leibler divergence between a proposed distribution and the true posterior distribution. We have assumed that the approximate posterior distribution is a Gaussian mixture. As a special case, we have found that when we choose the number of components in the Gaussian mixture as one, our solution reduces to the posterior linearization filter. Additionally, we have demonstrated that if we choose the number of components as one and assume that the measurements are linear, we obtain the Kalman filter update equations. In other words, we have proved that the Kalman filter also minimizes the Kullback-Leibler divergence. We have shown that the proposed solution is better than Gaussian filters and the standard Gaussian sum filter. In addition, we have also demonstrated that the proposed solution with a moderately low number of components performs similarly to the particle filter.

Our solution to the nonlinear filtering problem is preferable when the use of Monte Carlo methods is not desired. In order to eliminate the necessity of using Monte Carlo methods, we have used some approximations in the solution. The computational time of the suggested solution is low compared to PF if we have a high dimensional problem. The NEES values of the proposed method have shown that the method provides consistent estimates. Our method fills the gap in the literature between the research on Gaussian sum filtering and nonlinear filtering with divergence minimization. The only disadvantage of the method is that when we use a high number of components for the assumed Gaussian mixture, the computation time of the algorithm is larger than the PF especially when the problem is low dimensional.

We have focused on the Bayesian filtering/smoothing problem with unknown noise covariances after solving the nonlinear filtering problem. We have solved the problem for unknown process noise covariance and unknown measurement noise covariance separately. We know that running a filtering or smoothing algorithm with unknown covariances results in poor performance and even divergence in the solution. Our proposed methods solve this problem by estimating unknown covariances and using the estimation results in the algorithm. Simulation results have shown that the performance of our solutions is similar to the performance of the Rauch-Tung-Striebel smoother which knows the noise covariances exactly.

In the nonlinear filtering part, the measurement model is nonlinear. We can extend this nonlinearity to the system model as a future work. We use Newton's method to solve the optimization part of the problem for which we use constant step size. One can change this step size adaptively to decrease the number of iterations required for the convergence. In the solution, we have proposed an update algorithm for the weights, means and covariance of the assumed posterior. The covariance update can be performed by using the observed information matrix and the remaining parameters can be updated according to the proposed update algorithm. Instead of minimizing forward KL divergence, future work may focus on minimizing the $J$-divergence between the proposed distribution and the posterior distribution particularly for multimodal distributions. Furthermore, a solution can also be derived by minimizing the $\alpha$-divergence. Once if the solution is found, one can consider adjusting the value of $\alpha$ to switch between different divergences to optimize the performance.

The work on smoothing with unknown noise covariances examines linear systems. This work can be extended such that it covers nonlinear systems in a future study. In addition, instead of concentrating on the process noise covariance and measurement noise covariance separately, one can try to calculate/approximate the joint posterior for the state and both noise covariances simultaneously. In the solution of the problem, we have approximated the $t$-distribution with a scale Gaussian mixture with a constant number of components. To decrease the computational cost of the algorithm, one can consider adjusting the number of components based on the degrees of freedom parameter of the $t$-distribution. The number of components in the Gaussian mixture can be decreased if the degrees of freedom of the $t$-distribution is large lowering the computational load of the algorithm.

# REFERENCES

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

[2] T. Minka, "Divergence measures and message passing," tech. rep., Microsoft Research, 2005.

[3] M.-M. Deza and E. Deza, *Dictionary of distances*. Elsevier, 2006.

[4] A. H. Sayed, *Adaptive Filters*. John Wiley & Sons, 2008.

[5] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[6] A. K. Gupta and D. K. Nagar, *Matrix Variate Distributions*. CRC Press, 2000.

[7] B. Schwartz, S. Gannot, and E. A. P. Habets, "Online speech dereverberation using Kalman filter and EM algorithm," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 2, pp. 394–406, 2015.

[8] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. John Wiley & Sons, 2001.

[9] M. S. Grewal, A. P. Andrews, and C. G. Bartone, *Global Navigation Satellite Systems, Inertial Navigation, and Integration*. John Wiley & Sons, 2020.

[10] C. Wells, *The Kalman Filter in Finance*, vol. 32. Springer Science & Business Media, 1996.

[11] A. Gelb, *Applied Optimal Estimation*. MIT Press, 1974.

[12] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

[13] Á. F. García-Fernández, L. Svensson, M. R. Morelande, and S. Särkkä, "Posterior linearization filter: Principles and implementation using sigma points," *IEEE Trans. Signal Process.*, vol. 63, no. 20, pp. 5561–5573, 2015.

[14] M. Raitoharju, L. Svensson, Á. F. García-Fernández, and R. Piché, "Damped posterior linearization filter," *IEEE Signal Process. Lett.*, vol. 25, no. 4, pp. 536–540, 2018.

[15] K. Ito, "Gaussian filter for nonlinear filtering problems," in *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No. 00CH37187)*, vol. 2, pp. 1218–1223, IEEE, 2000.

[16] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Trans. Autom. Control*, vol. 54, no. 6, pp. 1254–1269, 2009.

[17] H. Sorenson and D. Alspach, "Recursive Bayesian estimation using Gaussian sums," *Automatica*, vol. 7, no. 4, pp. 465–479, 1971.

[18] D. Alspach and H. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Trans. Autom. Control*, vol. 17, no. 4, pp. 439–448, 1972.

[19] R. Bucy and K. Senne, "Digital synthesis of non-linear filters," *Automatica*, vol. 7, no. 3, pp. 287–298, 1971.

[20] J. Královec and M. Šimandl, "Filtering, prediction and smoothing with point-mass approach," *IFAC Proceedings Volumes*, vol. 37, no. 6, pp. 375–380, 2004.

[21] D. S. Kalogerias and A. P. Petropulu, "Grid based nonlinear filtering revisited: Recursive estimation & asymptotic optimality," *IEEE Trans. Signal Process.*, vol. 64, no. 16, pp. 4244–4259, 2016.

[22] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. F Radar Signal Process.*, vol. 140, no. 2, pp. 107–113, 1993.

[23] A. Doucet, A. Smith, N. de Freitas, and N. Gordon, eds., *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

[24] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, 2002.

[25] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Trans. Signal Process.*, vol. 50, pp. 425–437, Feb. 2002.

[26] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle filtering," *IEEE Signal Process. Mag.*, vol. 20, pp. 19–38, Sept. 2003.

[27] A. Doucet and X. Wang, "Monte Carlo methods for signal processing: A review in the statistical signal processing context," *IEEE Signal Process. Mag.*, vol. 22, pp. 152–170, Nov. 2005.

[28] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 25, pp. 53–82, July 2010.

[29] M. Raitoharju, S. Ali-Löytty, and R. Piché, "Binomial Gaussian mixture filter," *EURASIP J. Adv. Signal Process.*, vol. 2015, no. 1, pp. 1–18, 2015.

[30] F. Faubel, J. McDonough, and D. Klakow, "The split and merge unscented Gaussian mixture filter," *IEEE Signal Process. Lett.*, vol. 16, no. 9, pp. 786–789, 2009.

[31] M. F. Huber, "Adaptive Gaussian mixture filter based on statistical linearization," in *Proceedings of International Conference on Information Fusion*, pp. 1–8, 2011.

[32] M. Raitoharju and S. Ali-Loytty, "An adaptive derivative free method for Bayesian posterior approximation," *IEEE Signal Process. Lett.*, vol. 19, no. 2, pp. 87–90, 2012.

[33] M. L. Psiaki, J. R. Schoenberg, and I. T. Miller, "Gaussian sum reapproximation for use in a nonlinear filter," *J. Guid. Control Dyn.*, vol. 38, no. 2, pp. 292–303, 2015.

[34] R. Zanetti and K. Tuggle, “A novel Gaussian mixture approximation for non-linear estimation,” in *Proceedings of International Conference on Information Fusion (FUSION)*, pp. 1100–1106, 2018.

[35] K. Tuggle and R. Zanetti, “Automated splitting Gaussian mixture nonlinear measurement update,” *J. Guid. Control Dyn.*, vol. 41, no. 3, pp. 725–734, 2018.

[36] G. Terejanu, P. Singla, T. Singh, and P. D. Scott, “Adaptive Gaussian sum filter for nonlinear Bayesian estimation,” *IEEE Trans. Autom. Control*, vol. 56, no. 9, pp. 2151–2156, 2011.

[37] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, “The variational approximation for Bayesian inference,” *IEEE Signal Process. Mag.*, vol. 25, no. 6, pp. 131–146, 2008.

[38] T. P. Minka, “Expectation propagation for approximate Bayesian inference,” in *Proceedings of Conference on Uncertainty in Artificial Intelligence*, pp. 362–369, 2001.

[39] T. P. Minka, *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

[40] J. Vermaak, N. D. Lawrence, and P. Perez, “Variational inference for visual tracking,” in *Proceedings of Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 773–780, June 2003.

[41] Y. Hu, Q. Pan, Z. Guo, Z. Shi, and Z. Hu, “Iterative nonlinear Kalman filtering via variational evidence lower bound maximization,” in *Proceedings of International Conference on Information Fusion (FUSION)*, pp. 1–6, July 2019.

[42] J. E. Darling and K. J. DeMars, “Minimization of the Kullback–Leibler divergence for nonlinear estimation,” *J. Guid. Control Dyn.*, vol. 40, no. 7, pp. 1739–1748, 2017.

[43] S. Gultekin and J. Paisley, “Nonlinear Kalman filtering with divergence minimization,” *IEEE Trans. Signal Process.*, vol. 65, no. 23, pp. 6319–6331, 2017.

[44] Y. Luo, C. Guo, J. Zheng, and S. You, “A non-linear filtering algorithm based on alpha-divergence minimization,” *Sensors*, vol. 18, no. 10, p. 3217, 2018.

[45] M. R. Morelande and Á. F. García-Fernández, "Analysis of Kalman filter approximations for nonlinear measurements," *IEEE Trans. Signal Process.*, vol. 61, no. 22, pp. 5477–5484, 2013.

[46] M. Raitoharju, Ángel F. García-Fernández, and R. Piché, "Kullback–Leibler divergence approach to partitioned update Kalman filter," *Signal Process.*, vol. 130, pp. 289–298, 2017.

[47] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[48] R. L. Streit, *Poisson Point Processes: Imaging, Tracking, and Sensing*. Springer Science & Business Media, 2010.

[49] D. J. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

[50] M. I. Ribeiro, "Kalman and extended Kalman filters: Concept, derivation and properties," *Institute for Systems and Robotics*, vol. 43, 2004.

[51] M. A. Skoglund, G. Hendeby, and D. Axehill, "Extended Kalman filter modifications based on an optimization view point," in *Proceedings of International Conference on Information Fusion (FUSION)*, pp. 1856–1861, 2015.

[52] B. M. Bell and F. W. Cathey, "The iterated Kalman filter update as a Gauss-Newton method," *IEEE Trans. Autom. Control*, vol. 38, no. 2, pp. 294–297, 1993.

[53] G. Bonnet, "Transformations des signaux aléatoires a travers les systèmes non linéaires sans mémoire," *Annales des Télécommunications*, vol. 19, pp. 203–220, 1964.

[54] R. Price, "A useful theorem for nonlinear devices having Gaussian inputs," *IRE Transactions on Information Theory*, vol. 4, pp. 69–72, June 1958.

[55] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *International Conference on Machine Learning*, pp. 1278–1286, 2014.

[56] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Prentice Hall, 2000.

[57] S. Sangsuk-Iam and T. Bullock, "Analysis of discrete-time Kalman filtering under incorrect noise covariances," *IEEE Trans. Autom. Control*, vol. 35, no. 12, pp. 1304–1309, 1990.

[58] H. Heffes, "The effect of erroneous models on the Kalman filter response," *IEEE Trans. Autom. Control*, vol. 11, no. 3, pp. 541–543, 1966.

[59] R. H. Shumway and D. S. Stoffer, "An approach to time series smoothing and forecasting using the EM algorithm," *Journal of Time Series Analysis*, vol. 3, no. 4, pp. 253–264, 1982.

[60] Z. Ghahramani and G. E. Hinton, "Parameter estimation for linear dynamical systems," Tech. Rep. CRG-TR-96-2, University of Toronto, Department of Computer Science, CA, 1996.

[61] B. Zheng, P. Fu, B. Li, and X. Yuan, "A robust adaptive unscented Kalman filter for nonlinear estimation with uncertain noise covariance," *Sensors*, vol. 18, no. 3, p. 808, 2018.

[62] R. Dehghannasiri, X. Qian, and E. R. Dougherty, "Robust smoothing for state-space models with unknown noise statistics," in *Proceedings of Asilomar Conference on Signals, Systems, and Computers*, pp. 1024–1028, 2018.

[63] Y. Li and J. Li, "Robust adaptive Kalman filtering for target tracking with unknown observation noise," in *Chinese Control and Decision Conference (CCDC)*, pp. 2075–2080, 2012.

[64] J. Li, G. Yuan, and H. Duan, "Adaptive Kalman filter for SINS/GPS integration system with measurement noise uncertainty," in *IEEE International Conference on Unmanned Systems (ICUS)*, pp. 163–168, 2021.

[65] R. Mehra, "On the identification of variances and adaptive Kalman filtering," *IEEE Trans. Autom. Control*, vol. 15, no. 2, pp. 175–184, 1970.

[66] R. Mehra, "Approaches to adaptive filtering," *IEEE Trans. Autom. Control*, vol. 17, no. 5, pp. 693–698, 1972.

[67] F. Tronarp, T. Karvonen, and S. Särkkä, “Student’s t filters for noise scale estimation,” *IEEE Signal Process. Lett.*, vol. 26, no. 2, pp. 352–356, 2019.

[68] F. Zhu, C. Xue, Y. Huang, and Y. Zhang, “Robust Student’s t-based Rauch-Tung-Striebel smoother for non-stationary heavy-tailed measurement noises,” in *Chinese Control Conference (CCC)*, pp. 3071–3076, 2021.

[69] Y. Huang, Y. Zhang, N. Li, and J. Chambers, “A robust Gaussian approximate fixed-interval smoother for nonlinear systems with heavy-tailed process and measurement noises,” *IEEE Signal Process. Lett.*, vol. 23, no. 4, pp. 468–472, 2016.

[70] Y. Huang, Y. Zhang, N. Li, and J. Chambers, “A robust Gaussian approximate filter for nonlinear systems with heavy tailed measurement noises,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4209–4213, 2016.

[71] Y. Huang, G. Jia, B. Chen, and Y. Zhang, “A new robust Kalman filter with adaptive estimate of time-varying measurement bias,” *IEEE Signal Process. Lett.*, vol. 27, pp. 700–704, 2020.

[72] M. J. Beal, *Variational algorithms for approximate Bayesian inference*. University of London, University College London (United Kingdom), 2003.

[73] S. Sarkka and A. Nummenmaa, “Recursive noise adaptive Kalman filtering by variational Bayesian approximations,” *IEEE Trans. Autom. Control*, vol. 54, no. 3, pp. 596–600, 2009.

[74] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[75] W. Li and Y. Jia, “State estimation for jump Markov linear systems by variational Bayesian approximation,” *IET Control Theory & Applications*, vol. 6, no. 2, pp. 319–326, 2012.

[76] K. Li, S. Zhao, C. K. Ahn, and F. Liu, “State estimation for jump Markov nonlinear systems of unknown measurement data covariance,” *Journal of the Franklin Institute*, vol. 358, no. 2, pp. 1673–1691, 2021.

[77] S. Sarkka and J. Hartikainen, "Variational Bayesian adaptation of noise covariances in non-linear Kalman filtering." `https://arxiv.org/abs/arXiv:1302.0681v1`, 2013.

[78] G. Agamennoni, J. I. Nieto, and E. M. Nebot, "Approximate inference in state-space models with heavy-tailed noise," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5024–5037, 2012.

[79] T. Ardeshiri, E. Özkan, U. Orguner, and F. Gustafsson, "Approximate Bayesian smoothing with unknown process and measurement noise covariances," *IEEE Signal Process. Lett.*, vol. 22, no. 12, pp. 2450–2454, 2015.

[80] P. Dong, Z. Jing, H. Leung, and K. Shen, "Variational Bayesian adaptive cubature information filter based on Wishart distribution," *IEEE Trans. Autom. Control*, vol. 62, no. 11, pp. 6051–6057, 2017.

[81] Y. Huang, Y. Zhang, Z. Wu, N. Li, and J. Chambers, "A novel adaptive Kalman filter with inaccurate process and measurement noise covariance matrices," *IEEE Trans. Autom. Control*, vol. 63, no. 2, pp. 594–601, 2018.

[82] J. Ma, H. Lan, Z. Wang, X. Wang, Q. Pan, and B. Moran, "Improved adaptive Kalman filter with unknown process noise covariance," in *Proceedings of International Conference on Information Fusion (FUSION)*, pp. 1–5, 2018.

[83] Y. Huang, Y. Zhang, P. Shi, and J. Chambers, "Variational adaptive Kalman filter with Gaussian-inverse-Wishart mixture distribution," *IEEE Trans. Autom. Control*, vol. 66, no. 4, pp. 1786–1793, 2021.

[84] G. Chang, C. Chao, Q. Zhang, and S. Zhang, "Variational Bayesian adaptation of process noise covariance matrix in Kalman filtering," *Journal of the Franklin Institute*, vol. 358, 03 2021.

[85] X. Yu, J. Li, and J. Xu, "Nonlinear filtering in unknown measurement noise and target tracking system by variational Bayesian inference," *Aerospace Science and Technology*, vol. 84, pp. 37–55, 2019.

[86] X. Dong, L. Chisci, and Y. Cai, "An adaptive variational Bayesian filter for nonlinear multi-sensor systems with unknown noise statistics," *Signal Processing*, vol. 179, p. 107837, 2021.

[87] C. Xu, S. Zhao, Y. Ma, B. Huang, F. Liu, and X. Luan, “Sensor fault estimation in a probabilistic framework for industrial processes and its applications,” *IEEE Trans. Ind. Informat.*, vol. 18, no. 1, pp. 387–396, 2022.

[88] A. P. Sage and G. W. Husa, “Adaptive filtering with unknown prior statistics,” in *Joint Automatic Control Conference*, no. 7, pp. 760–769, 1969.

[89] Y. Shi, C. Han, and Y. Liang, “Adaptive UKF for target tracking with unknown process noise statistics,” in *Proceedings of International Conference on Information Fusion*, pp. 1815–1820, 2009.

[90] K. Berntorp and S. Di Cairano, “Approximate noise-adaptive filtering using Student-t distributions,” in *Proceedings of American Control Conference (ACC)*, pp. 2745–2750, 2018.

[91] A. El Attar, A. Pigeau, and M. Gelgon, “Fast aggregation of Student mixture models,” in *European Signal Processing Conference*, pp. 2638–2642, 2009.

[92] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization*. Cambridge university press, 2004.

## CURRICULUM VITAE

### PERSONAL INFORMATION

**Surname, Name:** Laz, Eray

### EDUCATION

| **Degree** | **Institution** | **Year of Graduation** |
|---|---|---|
| M.S. | Bilkent University, Electrical and Electronics Engineering | 2016 |
| B.S. | Bilkent University, Electrical and Electronics Engineering | 2013 |
| High School | Sami Yangın Anatolian High School | 2008 |

### PROFESSIONAL EXPERIENCE

| **Year** | **Place** | **Enrollment** |
|---|---|---|
| 2013-Present | ASELSAN, Inc. | Systems Engineer |
| 2012 June | HAVELSAN, Inc. | Intern Engineering Student |
| 2011 July | ASELSAN, Inc. | Intern Engineering Student |

**PUBLICATIONS**

1. E. Laz and U. Orguner, "Bayesian filtering and smoothing with unknown measurement noise covariance," *submitted to IEEE International Radar Conference*, 2023.

2. E. Laz and U. Orguner, "Gaussian mixture filtering with nonlinear measurements minimizing forward Kullback-Leibler divergence," *Signal Processing*, vol. 208, Article 108992, July 2023.

3. E. Laz, "Optimal cost allocation in centralized and decentralized detection systems using Bhattacharyya distance," in *2017 IEEE Radar Conference (RadarConf)*, pp. 1170–1173, May 2017.

4. E. Laz and S. Gezici, "Centralized and decentralized detection with cost-constrained measurements," *Signal Processing*, vol. 132, pp. 8–18, March 2017.

5. E. Laz and S. Gezici, "Optimal cost allocation in centralized and decentralized detection problems," in *2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, July 2016.