



**This electronic thesis or dissertation has been
downloaded from Explore Bristol Research,
<http://research-information.bristol.ac.uk>**

Author:

Hao, Pengcheng

Title:

Novel Computational Methods for State Space Filtering

General rights

Access to the thesis is subject to the Creative Commons Attribution - NonCommercial-No Derivatives 4.0 International Public License. A copy of this may be found at <https://creativecommons.org/licenses/by-nc-nd/4.0/legalcode>. This license sets out your rights and the restrictions that apply to your access to the thesis so it is important you read this before proceeding.

Take down policy

Some pages of this thesis may have been removed for copyright restrictions prior to having it been deposited in Explore Bristol Research. However, if you have discovered material within the thesis that you consider to be unlawful e.g. breaches of copyright (either yours or that of a third party) or any other law, including but not limited to those relating to patent, trademark, confidentiality, data protection, obscenity, defamation, libel, then please contact collections-metadata@bristol.ac.uk and include the following information in your message:

- Your contact details
- Bibliographic details for the item, including a URL
- An outline nature of the complaint

Your claim will be investigated and, where appropriate, the item in question will be removed from public view as soon as possible.

Novel Computational Methods for State Space Filtering

By

PENGCHENG HAO



Department of Electrical & Electronic Engineering
UNIVERSITY OF BRISTOL

A dissertation submitted to the University of Bristol
in accordance with the requirements of the degree of
DOCTOR OF PHILOSOPHY in the Faculty of Engineering.

AUGUST 2023

Word count: 19758

ABSTRACT

The state-space formulation for time-dependent models has been long used in various applications in science and engineering. While the classical Kalman filter (KF) provides optimal posterior estimation under linear Gaussian models, filtering in nonlinear and non-Gaussian environments remains challenging.

Based on the Monte Carlo approximation, the classical particle filter (PF) can provide more precise estimation under nonlinear non-Gaussian models. However, it suffers from particle degeneracy. Drawing from optimal transport theory, the stochastic map filter (SMF) accommodates a solution to this problem, but its performance is influenced by the limited flexibility of nonlinear map parameterisation. To account for these issues, a hybrid particle-stochastic map filter (PSMF) is first proposed in this thesis, where the two parts of the split likelihood are assimilated by the PF and SMF, respectively. Systematic resampling and smoothing are employed to alleviate the particle degeneracy caused by the PF. Furthermore, two PSMF variants based on the linear and nonlinear maps (PSMF-L and PSMF-NL) are proposed, and their filtering performance is compared with various benchmark filters under different nonlinear non-Gaussian models.

Although achieving accurate filtering results, the particle-based filters require expensive computations because of the large number of samples involved. Instead, robust Kalman filters (RKF) provide efficient solutions for the linear models with heavy-tailed noise, by adopting the recursive estimation framework of the KF. To exploit the stochastic characteristics of the noise, the use of heavy-tailed distributions which can fit various practical noises constitutes a viable solution. Hence, this thesis also introduces a novel RKF framework, RKF-SG α S, where the signal noise is assumed to be Gaussian and the heavy-tailed measurement noise is modelled by the sub-Gaussian α -stable (SG α S) distribution. The corresponding joint posterior distribution of the state vector and auxiliary random variables is estimated by the variational Bayesian (VB) approach. Four different minimum mean square error (MMSE) estimators of the scale function are presented. Besides, the RKF-SG α S is compared with the state-of-the-art RKFs under three kinds of heavy-tailed measurement noises, and the simulation results demonstrate its estimation accuracy and efficiency.

One notable limitation of the proposed RKF-SG α S is its reliance on precise model parameters, and substantial model errors can potentially impede its filtering performance. Therefore, this thesis also introduces a data-driven RKF method, referred to as RKFnet, which combines the conventional RKF framework with a deep learning technique. An unsupervised scheduled sampling technique (USS) is proposed to improve the

stability of the training process. Furthermore, the advantages of the proposed RKFnet are quantified with respect to various traditional RKFs.

DEDICATION AND ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to the following individuals and groups who have supported me throughout my journey in completing this thesis:

I would like to first to thanks to the Chinese Scholarship Council, who has funded my research work during this journey, and the University of Bristol, who has facilitated my research.

I am deeply thankful to my supervisor, Prof. Alin Achim, for his unwavering support, valuable insights, and continuous guidance throughout the entire process of conducting research and writing this thesis. His constructive feedback has been pivotal in refining my research approach and strengthening my academic skills.

I extend my heartfelt appreciation to my advisor, Dr. Oktay Karakus, for his invaluable advice. During the research process, his suggestions have expanded my knowledge on Bayesian theory and inspired many of my research topics.

I am grateful to my second supervisor, Dr. Pui Anantrasirichai, for her insightful feedback and assistance. She was always patient to listen to my research progress, answer my questions and express her advice on my projects.

I would like to acknowledge the feedback provided by my annual reviewer, Dr. Dimitris Agrafiotis. His thoughtful comments have contributed significantly to the refinement of my research process.

I extend my appreciation to my colleagues and friends who have been a constant source of support, encouragement, and camaraderie. Your discussions, brainstorming sessions, and shared experiences have made this academic endeavor enjoyable and rewarding. To Dr. Yanan Liu, for his generous shared research experience. To Dr. Yuming Hang, for his constant encouragement. To Dr. Bin Zhu, for his great advice on my research career. To Dr. Aaron Zhang, for his valuable suggestions on my work. To Haichen Zhou, for his advice on my mathematical formulation. To my dear teammates, Xin Tian, Ruixiong Wang, Ziyao Zhang, Xudong Ma, Tianqi Yang, our pleasure discussions were always helpful to my research. To many of my dear colleagues in Visual Information Laboratory, for your kind help on my research.

I am deeply indebted to my family for their unwavering love, encouragement, and belief in my abilities. Their constant support and understanding have been the cornerstone of my academic journey.

Lastly, I extend my gratitude to all those who have contributed to my growth as a researcher and scholar, directly or indirectly. Your support has been instrumental in making this accomplishment possible.

AUTHOR'S DECLARATION

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: DATE:

TABLE OF CONTENTS

	Page
List of Tables	xi
List of Figures	xiii
1 Introduction	1
1.1 Motivation	1
1.2 State-Space Model	2
1.3 Sequential Bayesian filtering and its challenges	3
1.4 Contributions, publications and outline	5
1.4.1 Contributions	5
1.4.2 Publications	6
1.4.3 Outline	7
2 Literature Review	9
2.1 Kalman filter and its extensions	9
2.1.1 Kalman filter	10
2.1.2 Extended Kalman filter	10
2.1.3 Unscented Kalman filter	11
2.2 Robust Kalman filters	12
2.2.1 M-estimator-based methods	13
2.2.2 Heavy-tailed-distribution-based methods	15
2.3 Particle filters	16
2.3.1 Importance sampling	16
2.3.2 Filtering framework	17
2.3.3 Resampling techniques	18
2.3.4 Importance proposal	19
2.4 Ensemble-based Kalman filters	20

TABLE OF CONTENTS

2.4.1	Ensemble Kalman filter	20
2.4.2	ESRF	21
2.5	Optimal-transport-based filters	22
2.5.1	Optimal transport fundamentals	23
2.5.2	OT-based filters	25
2.5.3	Stochastic Map Filtering	26
2.6	Hybrid filters	28
2.6.1	GMM-EnKF	29
2.6.2	SIR-ESRF	31
2.7	Data-driven filters	31
2.7.1	DNN fundamentals	32
2.7.2	RNN	33
2.7.3	DD filters	36
2.7.4	DD-MB filtering methods	36
2.8	Conclusion	38
3	A hybrid particle-stochastic map filter	41
3.1	Stochastic Map-Augmented Particle Filtering	41
3.2	Simulation Results	44
3.2.1	State-Space Models	44
3.2.2	Benchmark Filters	48
3.2.3	Performance Evaluation in Relation to Ensemble Size	49
3.2.4	The effect of θ on filter performance	54
3.2.5	The effect of the smoothing step on filter performance	59
3.2.6	Computational Complexity Analysis	60
3.3	Conclusion	61
4	Robust Kalman filters based on the sub-Gaussian α-stable distribution	65
4.1	α -stable Distributions and the Sub-Gaussian Case	66
4.2	RKF-SGaS	68
4.2.1	Proposed Robust Kalman Filtering Framework	69
4.2.2	MMSE Estimators of the Scale Function	72
4.2.3	Convergence Test for Fixed-point Iteration Method	77
4.3	Improved RKF-SL	77
4.4	Numerical simulations	79
4.4.1	Target Tracking Models and Noises	79

4.4.2	Benchmark Filters	79
4.4.3	Estimation of RKF Parameters	80
4.4.4	Performance Evaluation of RKF-SG α S Variants	80
4.4.5	RKF-SG α S vs. Benchmark Filters	86
4.5	Conclusion	88
5	RKFnet: a novel data-driven robust Kalman filter	89
5.1	New hierarchical Gaussian state-space model based on an unknown heavy-tailed distribution	89
5.2	Proposed RKFnet architecture	90
5.3	Unsupervised training algorithm	92
5.4	Simulation results	94
5.4.1	Target-tracking models and traditional RKFs	94
5.4.2	RKFnet parameters	94
5.4.3	Training dataset and optimiser	95
5.4.4	Performance evaluation in relation to p_{min} of USS	95
5.4.5	Comparison between the employed ST loss, L1 and L2	95
5.4.6	RKFnet vs traditional RKFs	95
5.5	Conclusion	97
6	Conclusion	101
6.1	Discussion and Future Work	102
	Bibliography	103

LIST OF TABLES

TABLES	Pages
3.1 RMSE results of the filters under different models. R and N represent RMSE and the ensemble size, respectively. M1:M5 represent UNGM, Henon-u, Henon-v, Lorenz-63, and Target tracking, respectively. F1:F9 represent PF, EnKF, ESRF, SMF-L, SMF-NL, SIR-ESRF, GMM-EnKF, PSMF-L, PSMF-NL. The first and second minimal RMSEs under different particle numbers and models are highlighted in red and blue bold fonts, respectively.	55
3.2 CRPS results of the filters under different models. C and N represent CRPS and the ensemble size, respectively. M1:M5 represent UNGM, Henon-u, Henon-v, Lorenz-63, and Target tracking, respectively. F1:F9 represent PF, EnKF, ESRF, SMF-L, SMF-NL, SIR-ESRF, GMM-EnKF, PSMF-L, PSMF-NL. The first and second minimal RMSEs under different particle numbers and models are highlighted in red and blue bold fonts, respectively.	56
4.1 Exemplary GSM distributions and their parameters	68
4.2 Parameter description of the mixing densities	68

LIST OF FIGURES

FIGURES	Pages
1.1 State-Space Model [33]	2
1.2 The procedure of sequential Bayesian filtering. At any time k , the observation assimilation includes the prediction and update steps [105].	3
2.1 Resampling methods [37]	18
2.2 (a) Monge transport map (b) Kantorovich's transport plan [53].	23
2.3 Deep neural network structure. (a) and (b) plot the forward pass and back-propagation steps, respectively.	32
2.4 RNN	33
2.5 vanilla RNN	34
2.6 LSTM	35
2.7 GRU	36
2.8 (a) and (b) show the original RNN-based filter and the PF-RNN-based filter, respectively.	37
2.9 The LSTM-KF-based filter.	37
2.10 KalmanNet	38
3.1 The comparison of the proposed PSMF to the traditional filters under different particle numbers. The horizontal axis represents the number of particles, and the vertical axis represents the $\log(\text{RMSE})$	50
3.2 The comparison of the proposed PSMF to the traditional filters under the different number of particles. The horizontal axis represents the number of particles, and the vertical axis represents the $\log(\text{CRPS})$	51
3.3 The comparison of the PSMF to the other hybrid filters under the different number of particles. The horizontal axis represents the number of particles, and the vertical axis represents $\log(\text{RMSE})$	52

LIST OF FIGURES

3.4	The comparison of the proposed PSMF to the traditional filters under the different number of particles. The horizontal axis represents the number of particles, and the vertical axis represents $\log(CRPS)$	53
3.5	The comparison among hybrid filters with different values of parameter θ under different models. The ensemble size N is set to 20. Horizontal axis represents the value of θ , and vertical axis represents the estimation $\log(RMSE)$	57
3.6	The comparison among hybrid filters with different values of parameter θ under different models. The ensemble size N is set to 200. Horizontal axis represents the value of θ , and vertical axis represents the estimation $\log(RMSE)$	58
3.7	The comparison among hybrid filters with different values of parameter θ under different models. The ensemble size N is set to 600. Horizontal axis represents the value of θ , and vertical axis represents the estimation $\log(RMSE)$	59
3.8	The comparison of the proposed PSMFs between with and without the smoothing step under the different number of particles. PSMF-nsm-L and PSMF-nsm-NL represent linear and nonlinear PSMFs without the smoothing step, respectively. The horizontal axis represents the number of particles, and the vertical axis represents $\log(RMSE)$	60
3.9	The comparison of the proposed PSMF to the other filters under the different number of particles. The horizontal axis represents the number of particles, and the vertical axis represents $\log(time)$	62
4.1	Mixing density and tail behaviour analysis. (a) and (b) plot the PDFs of the SGaS distribution and its mixing density, respectively. Also, the corresponding tail behaviour is shown in (c) and (d).	69
4.2	The target tracking results of the RKF-SGaS-IS under different particle numbers N . For (a), the x-axis is the particle number index and the y-axis represents the shape parameter value. In comparison, for (b)-(e), the x-axis is the shape parameter value, and the y-axis represents the estimation RMSE, iteration number and $\log(time)$, respectively. (a) depicts the shape parameter ranges, where the filter can track the target steadily. By contrast, (b) and (c) show the position and velocity estimation RMSEs, respectively. Also, (d) describes the average fixed-point iteration numbers. Besides, the implementation time is shown in (e).	81

- 4.3 The target tracking results of the RKF-SG α S-GLQ under different root numbers L . For (a), the x-axis is the root number index and the y-axis represents the shape parameter value. In comparison, for (b)-(e), the x-axis is the shape parameter value, and the y-axis represents the estimation RMSE, iteration number and $\log(\text{time})$, respectively. (a) depicts the shape parameter ranges, where the filter can track the target steadily. By contrast, (b) and (c) show the position and velocity estimation RMSEs, respectively. Also, (d) describes the average fixed-point iteration numbers. Besides, the implementation time is shown in (e). 82
- 4.4 The target tracking results of the RKF-SG α S-GSGL under different root numbers L . For (a), the x-axis is the root number index and the y-axis represents the shape parameter value. In comparison, for (b)-(f), the x-axis is the shape parameter value, and the y-axis represents the estimation RMSE, iteration number, $\log(\text{time})$ and ratio value, respectively. (a) depicts the shape parameter ranges, where the filter can track the target. By contrast, (b) and (c) shows the position and velocity estimation RMSEs, respectively. Also, (d) describes the average fixed-point iteration numbers. Then, the implementation time is shown in (e). Besides, (f) paints the ratio of the iterations where the GS method is replaced with the GLQ estimator. 83
- 4.5 The target tracking results of the RKF-SG α S-GSIS under different particle numbers N . For (a), the x-axis is the root number index and the y-axis represents the shape parameter value. In comparison, for (b)-(f), the x-axis is the shape parameter value, and the y-axis represents the estimation RMSE, iteration number, $\log(\text{time})$ and ratio value, respectively. (a) depicts the shape parameter ranges, where the filter can track the target. By contrast, (b) and (c) shows the position and velocity estimation RMSEs, respectively. Also, (d) describes the average fixed-point iteration numbers. Then, the implementation time is shown in (e). Besides, (f) depicts the ratio of the iterations where the GS method is replaced with the IS estimator. 84
- 4.6 The effective shape parameter ranges of different filters under the three kinds of measurement noises. The titles of the subplots are the noise distributions. Also, for simplicity, the parameter ρ of GM distribution is defined as the shape parameter. 86

LIST OF FIGURES

4.7	The position and velocity estimation RMSEs of different filters under the three kinds of measurement noises. The titles of the subplots follow the format: position/velocity-noise distributions. Also, for simplicity, the ρ of GM distribution is defined as the shape parameter.	87
4.8	The execution time of different filters under the three kinds of measurement noises. The titles of the subplots are the noise distributions. Also, for simplicity, the ρ of GM distribution is defined as the shape parameter.	88
5.1	Structure of the RKFnet	93
5.2	The unsupervised scheduled sampling technique.	94
5.3	The tracking results of the RKFnet under the RKF-SG α S measurement noise. In every subplot, the title is the shape parameter α of the SG α S distribution, the x-axis represents p_{min} , and the y-axis is the position estimation error. Also, the "MC" represents the 5 independent Monte Carlo experiment results, and the corresponding maximal and minimal errors are depicted by "Maximum" and "Minimum", respectively. The filtering performance of the RKF-SG α S-GSIS is shown as a reference.	96
5.4	The filtering results of the RKFnet with different loss functions. The titles of the subplots are the distributions of the heavy-tailed measurement noise. The x-axis is the shape parameter index, and the y-axis represents the position estimation error. (a) plots the estimation errors under the GM noise, while the results under the ST and SG α S noise are shown in (b) and (c), respectively. 97	
5.5	The filtering performance comparison between the RKFnet and traditional RSTKF, RKF-slash and RKF-SG α S. The titles of the subplots are the distributions of the heavy-tailed measurement noise. The x-axis is the shape parameter index, and the y-axis represents the position estimation error. (a) plots the estimation errors under the GM noise, while the results under the ST and SG α S noise are shown in (b) and (c), respectively.	98

LIST OF ALGORITHMS

ALGORITHMS	Pages
1 One Time Step of the Sequential Bayesian filtering	4
2 Kalman filtering framework	10
3 EKF framework	11
4 UKF framework	13
5 EPF framework	20
6 EnKF framework	21
7 ESRF framework	22
8 GMM-EnKF framework	30
9 SIR-ESRF framework	31
10 The PSMF Algorithm	45
11 GSIS and GSGL estimation	77
12 One Time Step of the Proposed RKF-SG α S	78

ACRONYMS

- CRPS** continuous ranked probability score. xi, 49, 52, 54, 56
- DD** data-driven. 7, 31, 32, 36, 38, 39
- DD-MB** data-model-driven. 6, 7, 32, 36, 97
- DMF** deterministic transport map filter. 63, 102
- DNN** deep neural network. 33
- DNNs** deep neural networks. 31
- dof** degree of freedom. 15, 16
- EKF** extended Kalman filter. 9–12, 20, 38
- EM** expectation–maximization. 68, 70, 80, 82
- EnKF** Ensemble Kalman filter. xi, 9, 20–22, 25, 28–30, 38, 39, 43, 44, 46, 48, 50, 54–56, 61
- EPF** extended particle filter. 19, 20
- ESRF** Ensemble Square Root filter. xi, 20–22, 28, 29, 43, 48, 50, 52–57, 61
- ESRFs** Ensemble Square Root filters. 20, 29, 31, 38, 39
- ESS** effective sample size. 2
- ETPF** ensemble transform particle filter. 25, 29, 43
- FCN** fully connected neural network. 6, 91, 92, 94, 97
- GCLT** generalized Central Limit Theorem. 66

- GLQ** Gauss-Laguerre quadrature. xv, 6, 72–74, 76, 80, 82–87
- GM** Gaussian Mixture. xv, xvi, 43, 44, 47, 79, 86–88, 95, 97, 98
- GMM-EnKF** Gaussian mixture model-EnKF hybrid filter. xi, 29, 30, 43, 48, 54–56, 61
- GRU** gated recurrent unit. 33, 35
- GS** gamma series. xv, 72, 74, 76, 80, 83–88
- GSGL** gamma series Gauss-Laguerre quadrature. xv, 6, 80, 83, 85, 86
- GSIS** gamma series importance sampling. xv, xvi, 6, 76, 80, 84–86, 88, 95, 96, 101
- GSM** Gaussian scale mixture. 67, 77
- HKF** Huber Kalman filter. 13, 14
- IS** importance sampling. xiv, xv, 6, 16, 17, 72–74, 76, 80–82, 84–88
- IW** inverse-Wishart. 70, 80, 90
- KF** Kalman filter. i, 1, 4, 5, 7, 9, 10, 12–16, 20, 21, 36, 38, 91
- KFTNCM** standard KF with true noise covariance matrices. 80
- KL** Kullback-Leibler. 25, 27
- KR** Knothe-Rosenblatt. 25–28
- LSTM** long short-term memory. 33–36
- MAP** maximum a posterior probability. 16, 77
- MB** model-based. 31, 32, 36, 38, 39
- MCKF** maximum correntropy Kalman filter. 14
- MCMC** Markov chain Monte Carlo. 68
- MLE** maximum likelihood estimation. 80
- MMSE** minimum mean square error. i, viii, 1, 4, 6, 9, 16, 65, 69, 72, 77, 78, 88, 101

- OT** optimal transport. 23–26, 28, 39
- PDF** probability density function. xiv, 15, 65, 68–70, 72, 74, 78, 90, 92
- PDFs** probability density functions. 5, 15
- PF** particle filter. i, xi, 2, 5, 6, 9, 16–20, 25, 28, 29, 31, 36, 38, 39, 41–43, 48–50, 55, 56, 59, 61, 101
- PSMF** particle-stochastic map filter. i, xi, xiii, xiv, xvii, 5–7, 29, 41, 42, 44, 45, 48–63, 65, 101, 102
- RBFs** radial basis functions. 26
- RKF** robust Kalman filter. i, ix, xiv–xvii, 2, 4–7, 9, 15, 38, 65, 67, 69, 76–91, 93–98, 101, 102
- RKF-GSM** robust Kalman filter based on the Gaussian scale mixture distributions. 16
- RKF-SL** RKF based on the slash distribution. viii, 65, 77, 80, 86, 95
- RKF-VG** RKF based on the variance Gamma distribution. 79, 80, 86
- RKFs** robust Kalman filters. i, ii, 2, 4, 6, 7, 12, 13, 15, 16, 38
- RMSE** root-mean-square error. xi, xiv–xvi, 49, 52, 54, 55, 81–85, 87
- RNN** recurrent neural network. xiii, 32–34, 36–38, 93
- RSTKF** robust Student’s t-based Kalman filter. xvi, 15, 80, 95, 98
- SG α S** sub-Gaussian α -stable. i, xiv–xvii, 6, 7, 65–69, 74–89, 95–98, 101, 102
- SMC** Sequential Monte Carlo. 5, 26
- SMF** stochastic map filter. i, xi, 5, 23, 26, 27, 41–43, 48, 50, 52, 55, 56, 59, 61, 101
- SSM** statistical similarity measure. 14, 15
- ST** Student’s t. xvi, 15, 79, 80, 86–88, 92, 95, 97, 98
- STF** ST-distribution-based filter. 15
- TVAR** time-varying autoregressive. 68

UKF unscented Kalman filte. 9, 11, 12, 20, 38

UPF unscented particle filter. 19

USS unsupervised scheduled sampling technique. i, 6, 7, 89, 93, 95, 97, 98, 102

UT unscented transform. 11, 12

VB variational Bayesian. i, 4, 5, 70, 96

VG variance Gamma. 80

INTRODUCTION

This chapter provides an introduction to this thesis. We begin by presenting the motivation for the work undertaken. Then, the description of the state-space model is provided in section 1.2, whilst the sequential Bayesian filtering and its challenges are discussed in section 1.3. The contributions, related publications and outline of this thesis are detailed in section 1.4.

1.1 Motivation

The state-space formulation for time-dependent models has been long used in various applications in science and engineering. For instance, in target tracking, the state vector represents the kinematics of the target [22, 40, 56, 63]; in weather prediction, it is related to temperature, pressure, humidity, etc [15, 21]; in oceanography, it could refer to the spatial pattern of surface currents [85, 106]; and in economics, it concerns interest rates, inflation [1, 66], etc. Sequential Bayesian filtering algorithms are essential for estimating the state of dynamic systems, incorporating noisy measurements and probabilistic models. When both the dynamic and observation models of the state-space model are linear and Gaussian, the minimum mean square error (MMSE) estimate can optimally be obtained by the Kalman filter (KF) [50]. By contrast, nonlinear and/or non-Gaussian situations pose challenges to filtering as closed-form solutions for the posterior distributions are not available, which inspires the development of various filters for different models in the literature. Specifically, for the linear model with heavy-tailed

noise, the robust Kalman filters (RKF) [39, 41, 104] provide efficient solutions. However, it remains challenging to find proper heavy-tailed distributions, which can both flexibly fit the practical noise and allow for a hierarchical Gaussian form required by the RKF framework. By contrast, the particle filter (PF) [6, 13, 17, 23] provides a Monte Carlo approximation solution for nonlinear non-Gaussian models. Nevertheless, it suffers from particle degeneracy, where a large proportion of the particles have negligible weights after a few filtering iterations and the effective sample size (ESS) decreases. Also, it is computationally expensive due to the use of numerous samples. Motivated by the limitations of the RKF and classical PF framework, this research aims to contribute to the design of more robust and efficient filters. The outcomes of this research have broad applications, spanning from autonomous navigation to financial forecasting and environmental monitoring. As the foundation of sequential Bayesian filtering, the next section will explain the formulation of the state-space model.

1.2 State-Space Model

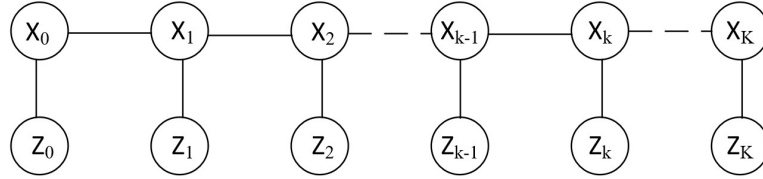


Figure 1.1: State-Space Model [33]

A state-space model can be described as a probabilistic graphical model, which defines the relationship between the hidden state variables and the measurements. It includes two equations, which are for states (signals) and their measurements. The signal model describes the state changes over time, whilst the measurement model explains the relationship between the states and measurements. A generic state-space model is expressed as

$$(1.1) \quad \begin{cases} \mathbf{x}_k = f_k(\mathbf{x}_{k-1}, \mathbf{w}_{k-1}) \\ \mathbf{z}_k = h_k(\mathbf{x}_k, \mathbf{v}_k) \end{cases}$$

where $\mathbf{x}_k \in \mathbb{R}^n$ and $\mathbf{z}_k \in \mathbb{R}^m$ denote the hidden state and the measurement vectors at time k , respectively. Also the process and measurement noises are represented as $\mathbf{w}_k \in \mathbb{R}^n$ and $\mathbf{v}_k \in \mathbb{R}^m$, respectively. The random model and observation noise errors are independent of

each other and the states. Besides, $f_k(\cdot)$ and $h_k(\cdot)$ are state transition and measurement functions. Figure 1.1 depicts the general conditionally independent structure of state-space models, which shows the independence of observations conditioned on the state. In this work, we only consider the static model with additive noise, and then the state-space model can be simplified to:

$$(1.2) \quad \begin{cases} \mathbf{x}_k = f(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1} \\ \mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \end{cases}$$

where $f(\cdot)$ and $h(\cdot)$ are static state transition and measurement functions. Finally, a linear discrete-time state-space model can be written as

$$(1.3) \quad \begin{cases} \mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \\ \mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \end{cases}$$

where $\mathbf{F} \in \mathbb{R}^{n \times n}$ and $\mathbf{H} \in \mathbb{R}^{m \times n}$ are the state transition and measurement matrices.

1.3 Sequential Bayesian filtering and its challenges

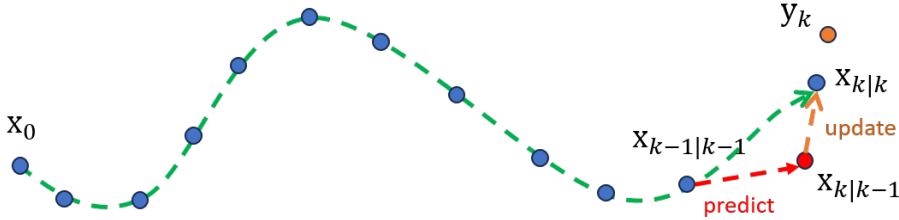


Figure 1.2: The procedure of sequential Bayesian filtering. At any time k , the observation assimilation includes the prediction and update steps [105].

Following the formulation of the state-space model, the sequential Bayesian filtering framework [96, 105] and its challenges are provided in this section. Sequential Bayesian filtering, also known as sequential Bayesian inference or estimation, is a statistical inference framework used to estimate the state of a stochastic process over time based on sequential observations. Specifically, at any time k , a sequential Bayesian filtering procedure contains two steps: prediction and update as shown in Figure 1.2. During the prediction step, the prior distribution $p(\mathbf{x}_k | \mathbf{z}_{1:k-1})$ is estimated. Assume the prior distribution $p(\mathbf{x}_0)$ is known, at any time k , $p(\mathbf{x}_k | \mathbf{z}_{1:k-1})$ can be estimated by

$$(1.4) \quad p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_{1:k-1}) p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1},$$

where $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ is the posterior distribution of the state at time $k-1$. According to the conditionally independent structure of the state-space model, we have

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_{1:k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1}).$$

Then the prior distribution $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ can be simplified as

$$(1.5) \quad p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1}.$$

During the update step, according to the Bayes' theorem [52], the posterior distribution $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ can be calculated by

$$(1.6) \quad p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{\int p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})d\mathbf{x}_k},$$

where the likelihood density $p(\mathbf{z}_k|\mathbf{x}_k)$ can be obtained based on the measurement model in (1.1). Then, the MMSE estimate of \mathbf{x}_k can be obtained by

$$(1.7) \quad \mathbf{E}(\mathbf{x}_k|\mathbf{z}_{1:k}) = \int \mathbf{x}_k p(\mathbf{x}_k|\mathbf{z}_{1:k})d\mathbf{x}_k.$$

The overall procedure of the sequential Bayesian filtering is shown in Algorithm 1.

Algorithm 1: One Time Step of the Sequential Bayesian filtering

Input: $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$

Prediction:

Estimate the prior distribution $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ based on equation (1.4);

Update:

Estimate the posterior distribution $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ based on equation (1.6);

return $p(\mathbf{x}_k|\mathbf{z}_{1:k})$

In the linear Gaussian scenario, the KF provides the optimal solution to $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. However, the presence of noise outliers can lead to suboptimal performance or even failure of the KF. To improve the filtering robustness, the RKF framework has been proposed, a non-Gaussian variant of the KF, specifically for linear models with heavy-tailed noise. The RKF combines the advantages of the recursive estimation framework of the KF with the M-estimators or heavy-tailed distributions. For the M-estimator-based RKFs, the Huber function [51, 94, 98], correntropy criterion [46, 64, 100] and statistical similarity measure [44, 45] have been employed. By contrast, an alternative RKF framework fits the model noise with the heavy-tailed distributions, enabling the subsequent approximation of the joint posterior distribution using the standard variational Bayesian (VB) [25, 95] approach. Due to the exploration of the stochastic properties of

the model noise, the RKF framework based on heavy-tailed distributions performs better than based on the M-estimators. However, most heavy-tailed distributions cannot be applied to the RKF framework as there are two requirements [41]. First, the distribution should exhibit flexible tail behaviour to properly fit various practical noises. Second, a hierarchical Gaussian form of the density should be available, so that the VB method under the RKF framework can be implemented. Due to these two conditions, only several heavy-tailed probability density functions (PDFs) have been utilised for the RKF framework and cannot fit various practical noises. Hence, investigating suitable heavy-tailed distributions for the RKF framework remains significant.

Despite their efficient filtering performance, both the KF and RKF frameworks rely on the linear assumption, limiting their estimation precision in highly nonlinear systems. By contrast, the PF, also known as the Sequential Monte Carlo (SMC) method, is the most popular filter for nonlinear non-Gaussian models and approximates the posterior distribution by a set of weighted particles. During the assimilation process, the samples are first propagated based on the signal model, and then their weights are updated based on the measurement model. However, there are two drawbacks of the PF framework. The first is particle degeneracy, where a small portion of particles have significant weights. This can hinder the effectiveness of the PF, particularly in scenarios with high-dimensional state spaces [75, 96] or complex nonlinear dynamics [6, 23]. Various techniques, such as effective importance proposals [96] and resampling strategies [37, 59], have been proposed to mitigate this problem. Also, compared with the KF and RKF, the PF is more computationally expensive due to its reliance on a substantial number of samples.

1.4 Contributions, publications and outline

1.4.1 Contributions

Building upon a comprehensive understanding of the challenges associated with the PF and RKF frameworks, three robust filters have been proposed and their filtering performance is evaluated for various state-space models and noises. The following are highlights of the novel contributions presented in this thesis:

1. A hybrid particle-stochastic map filter (PSMF) [33] was proposed, which combines the PF and the stochastic map filter (SMF) [87]. Specifically, the PSMF splits the likelihood into two parts, which are then updated by PF and SMF, respectively.

Also, systematic resampling [37, 59] and smoothing strategies are adopted to break the particle degeneracy caused by the PF. To investigate the influence of the nonlinearity of transport maps [53, 97], we introduce two variants of the proposed filter, the PSMF-L and PSMF-NL, which are based on linear and nonlinear maps, respectively. The filtering performance PSMF is validated under four nonlinear state-space models.

2. A robust Kalman filter based on the sub-Gaussian α -stable (SG α S) distribution [34, 55] was proposed in Chapter 4. Four different estimators are proposed to approximate the MMSE of the scale function. For the first two methods, the expectation of the scale function is approximated by the importance sampling (IS) [93] and Gauss-Laguerre quadrature (GLQ) methods [54], respectively. Two additional hybrid estimators, the gamma series importance sampling (GSIS) and gamma series Gauss-Laguerre quadrature (GSGL), are proposed for situations when the proposed GS estimator diverges and is thus replaced with the IS and GLQ-based methods, respectively. Besides, the RKF-SG α S is compared with the state-of-the-art RKFs in three types of heavy-tailed measurement noises, and the simulation results demonstrate its estimation accuracy and efficiency.
3. Chapter 5 presents a new state-space model where the unknown heavy-tailed distribution of the measurement noise is written in a hierarchical Gaussian form. Then, based on the proposed model, a novel data-model-driven (DD-MB) RKF framework called RKFnet is introduced, where the value of a mixing-parameter-based function is estimated by a fully connected neural network (FCN) [57] and the fixed scale matrix is directly calculated based on a trained neural network parameter. Also, to improve the convergence stability of the training process and overcome the difficulties in obtaining ground-truth data, an unsupervised scheduled sampling technique (USS) [8] is designed based on the filtering results from the traditional heavy-tailed-distribution-based RKFs. Besides, the influences of the USS parameters and the loss-function selection are evaluated in the experimental analysis. Finally, the performance of the proposed RKFnet is quantified in comparison with various traditional RKFs.

1.4.2 Publications

1. **Hao, P.**, Karakuş, O. and Achim, A. (2023) ‘A hybrid particle-stochastic map filter’, *Signal Processing*, 207, p. 108969. doi:10.1016/j.sigpro.2023.108969.

2. **Hao, P.**, Karakuş, O. and Achim, A. (2023) Robust kalman filters based on the sub-gaussian α -stable distribution, arXiv.org. Available at: <https://arxiv.org/abs/2305.07890>.

1.4.3 Outline

This manuscript is structured as follows:

Chapter 2: Literature review. This chapter provides a review related to our research work. First, the developments of the traditional filters are presented, including the KF, the RKF framework and various nonlinear filters. Also, data-driven (DD) filters [16, 28, 60] relying on the deep-learning technique are introduced.

Chapter 3: A hybrid particle-stochastic map filter. Following the introduction of the existing filters in Chapter 2, our proposed particle-based nonlinear filter, the PSMF, is explained in this chapter. We first provide the formulation of the PSMF, and then its filtering performance is shown under different benchmark models.

Chapter 4: Robust Kalman filters Based on the sub-Gaussian α -stable distribution. Considering the expensive computational cost of the PSMF, this chapter focuses on a new efficient RKF variant, the RKF-SG α S [34]. We start with the fundamentals of the SG α S distribution. Also, the theoretical explanation of this new RKF framework is provided. Besides, its estimation accuracy and efficiency are demonstrated in the target tracking scenarios with different heavy-tailed measurement noises.

Chapter 5: RKFnet: a novel data-driven robust Kalman filter. To alleviate the reliance of the traditional heavy-tailed-distribution-based RKFs on precise model parameters, Chapter 5 presents the DD-MB RKFnet. We first explain the new hierarchical Gaussian state-space model, and then the structure of the RKFnet and USS training method are detailed. Also, the comparison between the RKFnet and traditional RKFs is shown based on the target-tracking experiments.

Chapter 6: Conclusions. The final chapter gives a conclusion to all the contributions of this thesis, and then the potential future work is provided.

LITERATURE REVIEW

This chapter provides a review of previous work related to our proposed filters. We start with the explanation of the classical KF and its nonlinear variants in Section 2.1, and then the RKF framework is reviewed in Section 2.2. Also, various advanced nonlinear filters are introduced from Section 2.3 to Section 2.6, including the PF, the Ensemble Kalman filter (EnKF) [11, 24], optimal-transport-based filters [76, 87] and particle-based hybrid filters. Besides, filters based on deep learning are described in Section 2.7.

2.1 Kalman filter and its extensions

In the case of linear Gaussian signal and observation models, the MMSE estimator underpins the KF [50]. However, despite its optimality in the aforementioned scenario, the KF is not applicable when either model is nonlinear. Various nonlinear extensions of the KF have been proposed in the literature, such as the extended Kalman filter (EKF) [48] and the unscented Kalman filter (UKF) [49, 101]. In this section, we first introduce the classic KF in Subsection 2.1.1, and then the formulation of the EKF and UKF is provided.

2.1.1 Kalman filter

A linear state-space model has been provided in equation (1.3). Assume the signal and observation noises are subject to zero-mean Gaussian distributions, i.e., $\mathbf{w}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ and $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$, where \mathbf{Q} and \mathbf{R} are the covariance matrixes. Then, the optimal posterior estimation about state \mathbf{x}_k can be obtained by the KF algorithm, which is shown in Algorithm 2. At any time k , we have the posterior mean and covariance estimates, $\hat{\mathbf{x}}_{k-1|k-1}$ and $\mathbf{P}_{k-1|k-1}$, from the last time. During the prediction step, the mean and covariance of the prior distribution, $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$, are calculated. Also, $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ are obtained based on the Kalman gain \mathbf{K}_k during the update step.

Algorithm 2: Kalman filtering framework

Input: $\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}, \mathbf{z}_k$

Prediction:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}\hat{\mathbf{x}}_{k-1|k-1}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^T + \mathbf{Q}$$

Update:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1})$$

$$\mathbf{P}_{k|k} = (\mathbf{I}_n - \mathbf{K}_k\mathbf{H})\mathbf{P}_{k|k-1}$$

Kalman Gain:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^T (\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R})^{-1}$$

return $\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}$

2.1.2 Extended Kalman filter

The KF is renowned for its ability to provide optimal estimation in the case of linear and Gaussian models. However, its direct applicability is limited when dealing with practical nonlinear models. To relax the linear assumption, the EKF employs a first-order Taylor expansion to the nonlinear models, enabling the utilization of the KF framework within the linearised context. A nonlinear model with additive noise is shown in equation (1.2). Assume $\mathbf{F}_k = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1|k-1}}$ and $\mathbf{H}_k = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k-1}}$, then we can obtain the EKF filtering framework as shown in Algorithm 3. According to the Taylor series, the linearisation approximations are accurate only if the second and higher-order terms are negligible, and introduce significant biases or errors in many practical complex scenarios.

Algorithm 3: EKF framework**Input:** $\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}, \mathbf{z}_k$

Prediction:

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1})$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}$$

Update:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1}))$$

$$\mathbf{P}_{k|k} = (\mathbf{I}_n - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}$$

Kalman Gain:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$$

return $\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}$ **2.1.3 Unscented Kalman filter**

To reduce the approximation error in the EKF framework, the UKF replaces the model linearisation with the unscented transform (UT) [47], of which the formulation is explained in Subsection 2.1.3.1. Then, the UKF framework is presented in Subsection 2.1.3.2.

2.1.3.1 Unscented transform

The UT is a novel method for estimating the statistics of a random variable which undergoes a nonlinear transformation. It is founded on the intuition that it is easier to approximate a Gaussian distribution than to approximate an arbitrary nonlinear function or transformation. Although this method bears a superficial resemblance to Monte Carlo-type methods, there is a fundamental difference. The samples are not drawn at random but according to a deterministic algorithm. Compared with the linearisation in the EKF, the UT can provide more precise mean and covariance estimation considering higher-order moments [49]. The UT procedure is shown below:

1. Approximate a random vector $\mathbf{x} \in \mathbb{R}^n$, of which mean is $\bar{\mathbf{x}}$ and covariance is $\mathbf{P}^{\mathbf{xx}}$, by $2n + 1$ weighted points $\mathbf{x}^i, i = 1, \dots, n$ given by

$$\begin{aligned}
 \mathbf{x}^0 &= \bar{\mathbf{x}} & w_{\text{mean}}^0 &= \lambda_{\text{UT}} / (n + \lambda_{\text{UT}}) & w_{\text{cov}}^0 &= w_{\text{mean}}^0 + \beta_{\text{UT}} + 1 - \alpha_{\text{UT}}^2 \\
 \mathbf{x}^i &= \bar{\mathbf{x}} + \left(\sqrt{(n + \lambda_{\text{UT}}) \mathbf{P}^{\mathbf{xx}}} \right)_i & w_{\text{mean}}^i &= w_{\text{cov}}^i = \frac{1}{2(n + \lambda_{\text{UT}})} \\
 \mathbf{x}^{i+n} &= \bar{\mathbf{x}} - \left(\sqrt{(n + \lambda_{\text{UT}}) \mathbf{P}^{\mathbf{xx}}} \right)_i & w_{\text{mean}}^{i+n} &= w_{\text{cov}}^{i+n} = \frac{1}{2(n + \lambda_{\text{UT}})} \\
 \lambda_{\text{UT}} &= \alpha_{\text{UT}}^2 (n + \kappa_{\text{UT}}) - n
 \end{aligned}
 \tag{2.1}$$

where $\alpha_{\text{UT}}, \beta_{\text{UT}}, \kappa_{\text{UT}}$ are the sample spread parameter, prior knowledge parameter and scale parameter, respectively. Also, $\left(\sqrt{(n + \lambda_{\text{UT}})\mathbf{P}^{\mathbf{xx}}}\right)_i$ is the i -th row or column of the matrix square root of $(n + \lambda_{\text{UT}})\mathbf{P}^{\mathbf{xx}}$. Besides, w_{mean}^i and w_{cov}^i are the mean and covariance weights of the i -th point.

2. Instantiate each point through a nonlinear function f to yield the set of transformed sigma points,

$$(2.2) \quad \mathbf{y}^i = f[\mathbf{x}^i]$$

3. The mean of the transformed distribution can be estimated by the weighted average of the sigma points,

$$(2.3) \quad \bar{\mathbf{y}} = \sum_{i=0}^{2n} w_{\text{mean}}^i \mathbf{y}^i$$

4. The corresponding covariance can be approximated by the weighted outer product of the transformed points,

$$(2.4) \quad \mathbf{P}_{\mathbf{yy}} = \sum_{i=0}^{2n} w_{\text{cov}}^i (\mathbf{y}^i - \bar{\mathbf{y}})(\mathbf{y}^i - \bar{\mathbf{y}})^T$$

2.1.3.2 UKF framework

The UT provides an efficient way to estimate the mean and covariance in the nonlinear projection. The UKF framework combines the unscented transform with the KF and can produce more precise estimates than the EKF. At any time k , the assimilation step includes the prediction and update steps as shown in Algorithm 4, where the covariance matrices $\mathbf{P}_k^{\mathbf{xz}}$ and $\mathbf{P}_k^{\mathbf{zz}}$ are estimated based on the UT algorithm.

2.2 Robust Kalman filters

Both the EKF and UKF constitute improvements on the classical KF under mildly-nonlinear models. However, in many practical scenarios, their Gaussian assumption is violated, and then accurate estimation cannot be ensured. Considering the linear models with heavy-tailed noise, various efficient RKF's have been proposed utilising the advantages of the recursive estimation framework of the KF. In Subsection 2.2.1, the M-estimator-based RKF's are introduced. Also, the heavy-tailed-distribution-based RKF's are explained in Subsection 2.2.2.

Algorithm 4: UKF framework**Input:** $\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}, \mathbf{z}_k$

Prediction:

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2n} w_{\text{mean}}^i \mathbf{x}_{k|k-1}^i$$

$$\mathbf{P}_{k|k-1} = \sum_{i=0}^{2n} w_{\text{cov}}^i \left(\mathbf{x}_{k|k-1}^i - \hat{\mathbf{x}}_{k|k-1} \right) \left(\mathbf{x}_{k|k-1}^i - \hat{\mathbf{x}}_{k|k-1} \right)^T + \mathbf{Q}$$

Update:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})$$

$$\mathbf{P}_k = \mathbf{P}_{k|k-1} - \mathbf{K}_k (\mathbf{P}_k^{\text{zz}} + \mathbf{R}) \mathbf{K}_k^T$$

return $\hat{\mathbf{x}}_k, \mathbf{P}_k$

Sigma points generation:

$$\mathbf{x}_{k-1}^0 = \hat{\mathbf{x}}_{k-1|k-1}$$

$$\mathbf{x}_{k-1}^i = \hat{\mathbf{x}}_{k-1|k-1} + \left(\sqrt{(n + \lambda_{\text{UT}}) \mathbf{P}_{k-1|k-1}} \right)_i$$

$$\mathbf{x}_{k-1}^{i+n} = \hat{\mathbf{x}}_{k-1|k-1} - \left(\sqrt{(n + \lambda_{\text{UT}}) \mathbf{P}_{k-1|k-1}} \right)_i$$

Unscented transform:

$$\mathbf{x}_{k|k-1}^i = f_k(\mathbf{x}_{k-1}^i)$$

$$\mathbf{y}_{k|k-1}^i = h_k(\mathbf{x}_{k|k-1}^i)$$

$$\hat{\mathbf{z}}_{k|k-1} = \sum_{i=0}^{2n} w_{\text{mean}}^i \mathbf{z}_{k|k-1}^i$$

$$\mathbf{P}_k^{\text{xx}} = \sum_{i=0}^{2n} w_{\text{cov}}^i \left(\mathbf{x}_{k|k-1}^i - \hat{\mathbf{x}}_{k|k-1} \right) \left(\mathbf{x}_{k|k-1}^i - \hat{\mathbf{x}}_{k|k-1} \right)^T$$

$$\mathbf{P}_k^{\text{zz}} = \sum_{i=0}^{2n} w_{\text{cov}}^i \left(\mathbf{z}_{k|k-1}^i - \hat{\mathbf{z}}_{k|k-1} \right) \left(\mathbf{z}_{k|k-1}^i - \hat{\mathbf{z}}_{k|k-1} \right)^T$$

Kalman gain:

$$\mathbf{K}_k = \mathbf{P}_k^{\text{xz}} (\mathbf{P}_k^{\text{zz}} + \mathbf{R})^{-1}$$

2.2.1 M-estimator-based methods

For the classic KF, its posterior estimate can be derived based on the minimisation of a quadratic objective function [46], which can be expressed as

$$J = \frac{1}{2} (\mathbf{z}_k - \mathbf{H} \mathbf{x}_k)^T \mathbf{R}^{-1} (\mathbf{z}_k - \mathbf{H} \mathbf{x}_k) + \frac{1}{2} (\mathbf{x}_k - \mathbf{F} \hat{\mathbf{x}}_{k-1|k-1})^T \mathbf{P}_{k|k-1}^{-1} (\mathbf{x}_k - \mathbf{F} \hat{\mathbf{x}}_{k-1|k-1}).$$

The KF is optimal for Gaussian noise but sub-optimal for other distributed noise, as only the second-order information is used in the objective function. By contrast, there are three main types of M-estimator-based RKF, where different robust functions were introduced to construct the objective functions.

First, the Huber Kalman filter (HKF) [51, 94, 98] is the most famous extension of the M-estimator to the Kalman filter setting, where a generalized robust maximum likelihood estimate is achieved by minimizing the Huber-function-based cost function

depending on the measurement residual vector. A Huber function can be written as

$$f(x) = \begin{cases} 0.5x^2, & \text{for } x < \delta_{\text{Huber}} \\ \delta_{\text{Huber}}|x| - 0.5\delta_{\text{Huber}}^2, & \text{for } x \geq \delta_{\text{Huber}} \end{cases}$$

where δ_{Huber} is a scale parameter. The Huber function is a mixture of the L1 and L2 norm and assigns a lower weight to outliers, which makes the HKF more robust to heavy-tailed noise.

The maximum correntropy Kalman filter (MCKF) [46, 64, 100] is an alternative method to handle state and measurement outliers. The cross-correntropy of two scalar random variables, x and y , is defined as

$$C_\sigma(x, y) = \mathbb{E}[\kappa_\sigma(x, y)] = \iint \kappa_\sigma(x, y)p(x, y)dx dy$$

where $\kappa_\sigma(x, y)$ denotes a positive definite kernel function satisfying Mercer's theory [14] and $p(x, y)$ is the joint distribution of variables x and y . Under the MCKF framework, $\kappa_\sigma(x, y)$ is selected as the Gaussian kernel function, which can be written as

$$\kappa_\sigma(x, y) = G_\sigma(\varepsilon) = \exp\left(-\frac{\varepsilon^2}{2\sigma^2}\right).$$

where $\varepsilon = x - y$ and $\sigma > 0$ is the kernel bandwidth. Then, we have Taylor series expansion of C_σ

$$C_\sigma(x, y) = \frac{1}{\sqrt{2\pi}\sigma} \sum_{\xi=0}^{\infty} \frac{(-1)^\xi}{2^\xi \sigma^{2\xi} \xi!} \mathbb{E}\left[(x - y)^{2\xi}\right]$$

Compared with the quadratic function of the KF, this correntropy criterion uses higher-order moment information. Following this, we have the loss function of the MCKF:

$$J = G_\sigma(\|\mathbf{z}_k - \mathbf{H}\mathbf{x}_k\|_{\mathbf{R}^{-1}}) + G_\sigma(\|\mathbf{x}_k - \mathbf{F}\hat{\mathbf{x}}_{k-1|k-1}\|_{\mathbf{P}_{k|k-1}^{-1}})$$

where

$$\begin{aligned} \|\mathbf{z}_k - \mathbf{H}\mathbf{x}_k\|_{\mathbf{R}^{-1}} &= \sqrt{(\mathbf{z}_k - \mathbf{H}\mathbf{x}_k)^T \mathbf{R}^{-1} (\mathbf{z}_k - \mathbf{H}\mathbf{x}_k)} \\ \|\mathbf{x}_k - \mathbf{F}\hat{\mathbf{x}}_{k-1|k-1}\|_{\mathbf{P}_{k|k-1}^{-1}} &= \sqrt{(\mathbf{x}_k - \mathbf{F}\hat{\mathbf{x}}_{k-1|k-1})^T \mathbf{P}_{k|k-1}^{-1} (\mathbf{x}_k - \mathbf{F}\hat{\mathbf{x}}_{k-1|k-1})} \end{aligned}$$

Third, in [44, 45], a statistical similarity measure (SSM) is introduced to quantify the similarity between two random vectors and then used to develop a novel outlier-robust Kalman filter. The SSM is defined as follows:

$$s(\mathbf{x}, \mathbf{y}) = \mathbb{E}[f(\|\mathbf{x} - \mathbf{y}\|^2)] = \iint f(\|\mathbf{x} - \mathbf{y}\|^2) p(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}$$

Several similarity functions are shown in [44]. Under the new RKF framework, lower bounds of the SSM between the state vector and the predicted state vector and between the measurement vector and the predicted measurement vector are maximized, and the posterior probability density function (PDF) is approximated as Gaussian, i.e.,

$$J = \operatorname{argmax} \left\{ \int f_x \left(\|\mathbf{z}_k - \mathbf{H}\mathbf{x}_k\|_{\mathbf{R}^{-1}}^2 \right) q(\mathbf{x}_k) d\mathbf{x}_k + \int f_z \left(\|\mathbf{x}_k - \mathbf{F}\hat{\mathbf{x}}_{k-1|k-1}\|_{\mathbf{P}_{k|k-1}^{-1}}^2 \right) q(\mathbf{x}_k) d\mathbf{x}_k \right\}$$

where f_x and f_z denote the similarity functions of state and measurement models, respectively. Also, $q(\mathbf{x}_k)$ is the approximated Gaussian posterior distribution of \mathbf{x}_k .

Owing to the various robust functions, the M-estimator-based RKFs can perform better than the KF under heavy-tailed noise. However, their filtering performance is limited as the stochastic properties of the heavy-tailed noise are not exploited.

2.2.2 Heavy-tailed-distribution-based methods

Different from the M-estimator-based RKFs, the heavy-tailed-distribution-based methods improve the filtering performance by fitting the noise with different probability densities. The Student's t (ST) distribution was the first model to be used for this purpose, and there were two types of approaches involving it. First, the ST-distribution-based RKFs assume that the measurements are contaminated by heavy-tailed noise. For example, in [4, 107], the symmetric measurement noise is modelled as an ST distribution. Also, skewed noise is fitted by the skewed ST distribution in [72]. However, when the signal noise is heavy-tailed, these filters are not applicable because of the Gaussian assumption which is violated. In a second group of techniques, RKFs are based on the assumption that both the one-step prediction and likelihood densities can be captured by ST distributions. In [38, 81], the ST-distribution-based filter (STF) considers that the heavy-tailed posterior PDF follows the ST distribution. However, its filtering accuracy is limited due to the requirement that the prediction and observation PDFs should be characterised by the same degree of impulsiveness. By contrast, in [39], the robust Student's t-based Kalman filter (RSTKF) eliminates this requirement and expresses the prediction and likelihood PDFs in hierarchical Gaussian forms. Hence, it exhibits higher computation complexity but produces more precise estimates. Despite all these improvements, two drawbacks of ST-distribution-based RKFs remain. First, they provide a rough estimation when the noise is skewed. Also, the degree of freedom (dof) value of the ST distribution is set manually for each filtering scenario, which degrades its overall performance.

Owing to these disadvantages, other heavy-tailed distributions have also been investigated to design RKFs. In [67, 99], an RKF based on the multivariate Laplace distribution

was considered. As the Laplace distribution is only determined by its mean and covariance, the selection of the dof value is unnecessary. Nevertheless, without a shape parameter, the filter cannot adapt to varying types of noise. Hence, for heavy-tailed and/or skewed state and measurement noises, the robust Kalman filter based on the Gaussian scale mixture distributions (RKF-GSM) was proposed in [41]. However, the MMSE estimate of the scale function is not available, and the maximum a posterior probability (MAP) estimate introduces more errors. Also, the dof values are still manually set, which debases the estimation precision.

2.3 Particle filters

Both the classic KF and the RKF's exhibit efficiency in certain scenarios. However, their filtering performance tends to degrade significantly when the state-space model becomes highly nonlinear. By contrast, the PF [6, 17, 23] excels in providing precise estimation under such nonlinear conditions. In Subsection 2.3.1, we introduce the IS technique, which is the basement of the PF. Also, the formulation of the PF framework is explained in Subsection 2.3.2. As two significant solutions to alleviate particle degeneracy, the resampling methods and efficient importance proposals [96] are presented in Subsection 2.3.3 and 2.3.4, respectively.

2.3.1 Importance sampling

The IS method is a Monte Carlo simulation technique used to estimate the properties of a target distribution by drawing samples from a proposal distribution, which is easier to generate samples from. For instance, for the expectation estimation of a function $f(\mathbf{x})$ where $\mathbf{x} \sim p(\mathbf{x})$, we have

$$(2.5) \quad \mathbb{E}[f(\mathbf{x})] = \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x} \approx \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}^i)$$

where \mathbf{x}^i is the i -th sample of $p(\mathbf{x})$ and N is the sample size. If $p(\mathbf{x})$ cannot be sampled directly, an importance proposal density $q(\mathbf{x})$ can be introduced, and $\mathbb{E}[f(\mathbf{x})]$ can be approximated by

$$(2.6) \quad \mathbb{E}[f(\mathbf{x})] = \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x} = \int f(\mathbf{x})\frac{p(\mathbf{x})}{q(\mathbf{x})}q(\mathbf{x})d\mathbf{x} \approx \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}^i)\frac{p(\mathbf{x}^i)}{q(\mathbf{x}^i)}$$

where \mathbf{x}^i is the i -th sample of $q(\mathbf{x})$.

2.3.2 Filtering framework

Based on the IS method in the last section, the filtering framework is detailed in this section. At any time k , the posterior distribution can be expressed as

$$(2.7) \quad \begin{aligned} p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) &= \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})}p(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}) \\ &\propto p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}). \end{aligned}$$

For nonlinear models, $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ has a complex form and cannot be directly sampled. Instead, the PFs introduce a proper importance proposal, and $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ can be approximated by the weighted particles based on the IS technique. Specifically, select an importance proposal $q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$, and then the posterior distribution can be approximated as

$$(2.8) \quad p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \approx \sum_{i=1}^N w_k^i \delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^i).$$

where $\mathbf{x}_{0:k}^i$ is the i -th sample from $q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ and its corresponding weight w_k^i can be expressed as

$$(2.9) \quad w_k^i \propto \frac{p(\mathbf{x}_{0:k}^i|\mathbf{z}_{1:k})}{q(\mathbf{x}_{0:k}^i|\mathbf{z}_{1:k})}.$$

For simplicity, the importance proposal is chosen to be factorable, i.e.,

$$(2.10) \quad q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) = q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k})q(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}).$$

Then, according to equations (2.7) and (2.10), the particle weights can be updated sequentially, i.e.,

$$(2.11) \quad \begin{aligned} w_k^i &\propto \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)p(\mathbf{x}_{0:k-1}^i|\mathbf{z}_{1:k-1})}{q(\mathbf{x}_k^i|\mathbf{x}_{0:k-1}^i, \mathbf{z}_{1:k})q(\mathbf{x}_{0:k-1}^i|\mathbf{z}_{1:k-1})} \\ &= w_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{0:k-1}^i, \mathbf{z}_{1:k})} \end{aligned}$$

Further, if $q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k}) = q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_k)$, the importance proposal and weight update equation can be simplified as

$$(2.12) \quad q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) = q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_k)q(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}),$$

$$(2.13) \quad w_k^i \propto w_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k)},$$

respectively. For the standard PF formulation, the importance proposal is chosen as

$$(2.14) \quad q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k) = p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i),$$

which relies only on the prior information and leads to the weight-updating equation as

$$(2.15) \quad w_k^i \propto w_{k-1}^i p(\mathbf{z}_k | \mathbf{x}_k^i).$$

As mentioned in the above sections, a common problem in particle filtering applications is the degeneracy phenomenon, where a large part of the particles will have negligible weights after a few iterations. It has been proved that the variance of the importance weights can only increase over time [9, 23], and thus it is impossible to avoid the degeneracy phenomenon. As an efficient method of mitigating particle degeneracy, the resampling techniques are introduced in the next subsection.

2.3.3 Resampling techniques

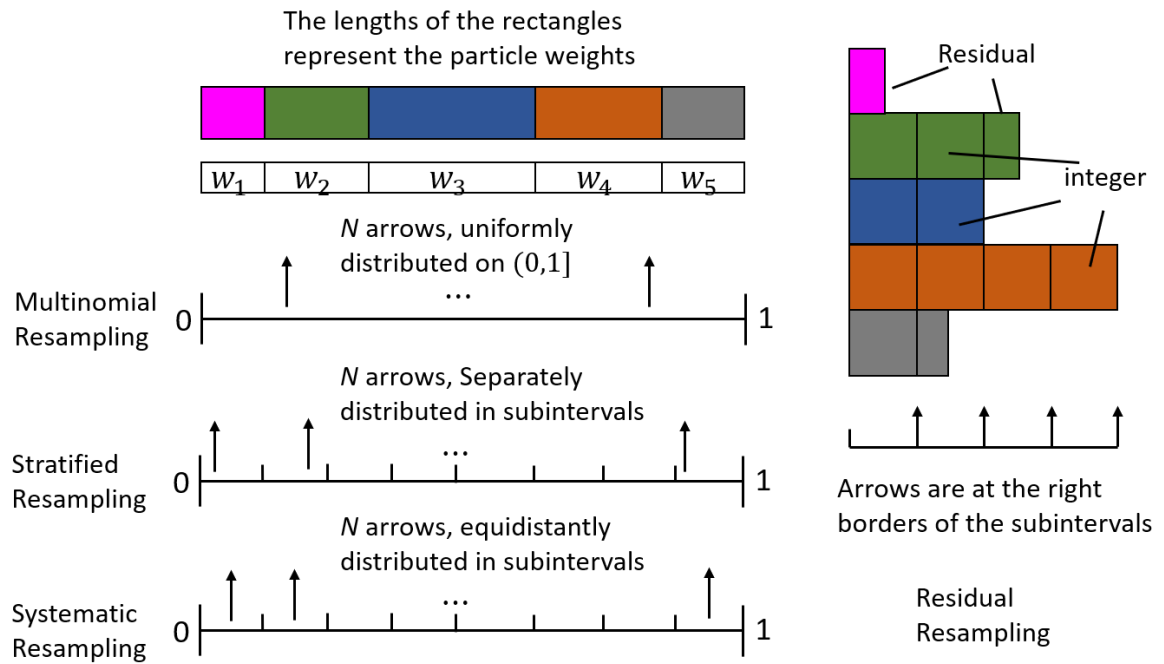


Figure 2.1: Resampling methods [37]

The resampling methods [37, 59] are used to statistically replicate high-weight particles and eliminate low-weight particles. In this section, we focus on four popular resampling methods: multinomial resampling, stratified Resampling, systematic

resampling and Residual resampling, as shown in Figure 2.1. Given samples $\mathbf{x}^i, i = 1 : N$ and their corresponding weights w^i , the cumulative probability distribution F of the normalised particle weights can be represented by

$$F(i) = \sum_{s=1}^{i-1} w^s$$

For the multinomial resampling, N ordered random numbers $u_i, i = 1 : N$ are produced by

$$(2.16) \quad u_i = \tilde{u}_i, \tilde{u}_i \sim U[0, 1)$$

where U represents the uniform distribution. Then the resampling particles can be obtained by

$$(2.17) \quad \mathbf{x}^i = \mathbf{x}(F^{-1}(u_i)), \text{ s.t. } u_i \in \left[\sum_{s=1}^{i-1} w^s, \sum_{s=1}^i w^s \right]$$

where function $\mathbf{x}(i) = \mathbf{x}^i$ and F^{-1} denotes the inverse of F . By contrast, the stratified resampling calculates u_i based on

$$(2.18) \quad u_i = \frac{(i-1) + \tilde{u}_i}{N}, \tilde{u}_i \sim U[0, 1).$$

Also, the systematic resampling generates u_i by

$$(2.19) \quad u_i = \frac{(i-1) + \tilde{u}}{N}, \tilde{u} \sim U[0, 1).$$

Besides, the Residual resampling first produces $\lfloor Nw_i \rfloor$ copies of \mathbf{x}^i and then employs one of the first three resampling algorithms to produce $N - \sum \lfloor Nw_i \rfloor$ particles based on the residual particle weights, $w'_i \propto Nw_i - \lfloor Nw_i \rfloor$.

2.3.4 Importance proposal

This section focus on an alternative method for the alleviation of particle degeneracy, importance proposals. For the standard PF, the importance proposal is only based on the prior information. In highly-nonlinear scenarios, the difference between the prior and posterior distributions causes serious particle degeneracy. To alleviate this problem, various advanced importance proposals have been explored by incorporating both the prior and observation information. For example, the extended particle filter (EPF) [5] and the unscented particle filter (UPF) [32] employ the importance proposals based on

the EKF and UKF, respectively. Also, the weighted ensemble Kalman filter [73] designs the importance proposal based on the EnKF detailed in Section 2.4. More importance proposals of the PFs can be found in the review paper [96]. In this section, we provide the assimilation procedure of the EPF in Algorithm 5 as an example.

Algorithm 5: EPF framework

Input: $\mathbf{x}_{k-1}^i, \mathbf{P}_{k-1|k-1}^i, \mathbf{z}_k$
 Design importance proposal by EKF:
 $\left(\hat{\mathbf{x}}_{k|k}^i, \mathbf{P}_{k|k}^i \right) = \mathbf{EKF} \left(\mathbf{x}_{k-1}^i, \mathbf{P}_{k-1|k-1}^i, \mathbf{z}_k \right)$
 $q \left(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k \right) = \mathcal{N} \left(\hat{\mathbf{x}}_{k|k}^i, \mathbf{P}_{k|k}^i \right)$
 Sampling from the importance proposal:
 $\mathbf{x}_k^i \sim \mathcal{N} \left(\hat{\mathbf{x}}_{k|k}^i, \mathbf{P}_{k|k}^i \right);$
 Update particle weights based on (2.11);
 Normalize sample weights and resampling;
 Estimate: $\hat{\mathbf{x}}_k = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_k^i$
return $\hat{\mathbf{x}}_k, \mathbf{x}_k^i, \mathbf{P}_{k|k}^i$

2.4 Ensemble-based Kalman filters

Both the resampling methods and importance proposals can alleviate the particle degeneracy, but this problem is still serious under high-dimensional and/or highly nonlinear models. To overcome this problem radically, the Ensemble-based Kalman filters [11, 24, 73] have been proposed as another solution to filtering under nonlinear models. There are two kinds of ensemble-based Kalman filters: the EnKF and the Ensemble Square Root filters (ESRFs) [92]. In particular, for EnKF, the linear transformation is performed statistically by treating the observations as random variables. By contrast, in ESRF, analysis perturbations follow the KF analysis error covariance equation. Different from the PF, both EnKF and ESRF just require a small number of particles as they perform posterior estimation through linear particle movements and then the particle weights are always equal.

2.4.1 Ensemble Kalman filter

In this section, we detail the EnKF framework, where the forecast step and analysis step are required at any time k . In the forecast step, each ensemble member is propagated

forward using the state-space model, and the forecast samples $\mathbf{x}_{k|k-1}^i$ and $\mathbf{z}_{k|k-1}^i$, $i = 1, \dots, N$ are produced by

$$\begin{aligned}\mathbf{x}_{k|k-1}^i &= f(\mathbf{x}_{k-1|k-1}^i) + \mathbf{w}_{k-1}^i \\ \mathbf{z}_{k|k-1}^i &= h(\mathbf{x}_{k|k-1}^i)\end{aligned}$$

where $\mathbf{x}_{k-1|k-1}^i$ is the i -th posterior sample at time $k-1$ and $\mathbf{w}_{k-1}^i \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$. By contrast, the analysis update aims to improve the ensemble's representation of the true state by combining the model predictions with the observational information, which is achieved by

$$\mathbf{x}_{k|k}^i = \mathbf{x}_{k|k-1}^i + \mathbf{K}_k (\mathbf{z}_k + \mathbf{v}_k^i - \mathbf{z}_{k|k-1}^i)$$

where $\mathbf{x}_{k|k}^i$ is the i -th posterior sample at time k , \mathbf{K}_k is the approximated Kalman gain and $\mathbf{v}_k^i \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$. The formulation of the EnKF framework is summarized in Algorithm 6.

Algorithm 6: EnKF framework

Input: $\mathbf{x}_{k-1|k-1}^i$

Prediction:

$$\begin{aligned}\mathbf{x}_{k|k-1}^i &= f(\mathbf{x}_{k-1|k-1}^i) + \mathbf{w}_{k-1}^i \\ \mathbf{z}_{k|k-1}^i &= h(\mathbf{x}_{k|k-1}^i)\end{aligned}$$

Update:

$$\mathbf{x}_{k|k}^i = \mathbf{x}_{k|k-1}^i + \mathbf{K}_k (\mathbf{z}_k + \mathbf{v}_k^i - \mathbf{z}_{k|k-1}^i)$$

Kalman Gain:

$$\begin{aligned}\mathbf{K}_k &= \mathbf{P}_k^{\mathbf{xz}} (\mathbf{P}_k^{\mathbf{zz}} + \mathbf{R})^{-1} \\ \mathbf{P}_k^{\mathbf{xz}} &= \frac{1}{N-1} \sum_{i=1}^N \left(\mathbf{x}_{k|k-1}^i - \bar{\mathbf{x}}_{k|k-1} \right) \left(\mathbf{z}_{k|k-1}^i - \bar{\mathbf{z}}_{k|k-1} \right)^T \\ \mathbf{P}_k^{\mathbf{zz}} &= \frac{1}{N-1} \sum_{i=1}^N \left(\mathbf{z}_{k|k-1}^i - \bar{\mathbf{z}}_{k|k-1} \right) \left(\mathbf{z}_{k|k-1}^i - \bar{\mathbf{z}}_{k|k-1} \right)^T \\ \bar{\mathbf{x}}_{k|k-1} &= \frac{1}{N} \sum_{i=1}^N \mathbf{x}_{k|k-1}^i \\ \bar{\mathbf{z}}_{k|k-1} &= \frac{1}{N} \sum_{i=1}^N \mathbf{z}_{k|k-1}^i\end{aligned}$$

return $\mathbf{x}_{k|k}^i$

2.4.2 ESRF

Compared with the KF, the EnKF provides a more precise posterior estimation under nonlinear non-Gaussian models. However, the perturbed observations introduce extra errors, especially when the ensemble size is small. To overcome this drawback, the ESRF

assimilates the observations based on the square root of the analysis error covariance matrix. As there are different square root solutions, various ESRF variants have been presented and were reviewed in [92]. In this section, we detail the version in [31], which is compared with our proposed filter in Chapter 3. First, the scaled ensemble perturbation matrix is calculated by

$$(2.20) \quad \mathbf{A} = \frac{1}{\sqrt{N-1}} \left[\mathbf{x}_{k|k-1}^1 - \bar{\mathbf{x}}_{k|k-1}, \dots, \mathbf{x}_{k|k-1}^N - \bar{\mathbf{x}}_{k|k-1} \right]$$

Assume observations are linear, and a single scalar observation z takes the form

$$(2.21) \quad z = \mathbf{h}^T \mathbf{x}_k + v$$

where \mathbf{h}^T extracts the observation z for the state vector \mathbf{x}_k and $v \sim \mathcal{N}(0, r^2)$. When the length of the observation vector is larger than one, the measurements can be assimilated sequentially. Following equations 2.20 and 2.21, the updated ensemble mean and perturbation matrix can be expressed as

$$\begin{aligned} \bar{\mathbf{x}}_{k|k} &= \bar{\mathbf{x}}_{k|k-1} + \frac{z - \mathbf{h}^T \bar{\mathbf{x}}_{k|k-1}}{\sigma^2 + r^2} \mathbf{A} \mathbf{v} \\ \mathbf{A}^a &= \mathbf{A} - b \mathbf{A} \mathbf{v} \mathbf{v}^T \end{aligned}$$

where $\mathbf{v}^T = \mathbf{h}^T \mathbf{A}$, $\sigma^2 = \mathbf{v}^T \mathbf{v}$ and $b = \frac{1}{\sigma^2 + r^2 + r \sqrt{\sigma^2 + r^2}}$. The updated ensemble can be calculated by

$$\mathbf{x}_{k|k}^i = \bar{\mathbf{x}}_{k|k} + \sqrt{N-1} \mathbf{A}_i^a$$

where \mathbf{A}_i^a is the i -th column of \mathbf{A}^a . The procedure of the ESRF is shown in Algorithm 7.

Algorithm 7: ESRF framework

Input: $\mathbf{x}_{k-1|k-1}^i$
 Prediction:
 $\mathbf{x}_{k|k-1}^i = f(\mathbf{x}_{k-1|k-1}^i) + \mathbf{w}_{k-1}^i$
 Update:
 $\mathbf{x}_{k|k}^i = \bar{\mathbf{x}}_{k|k} + \sqrt{N-1} \mathbf{A}_i^a$
return $\mathbf{x}_{k|k}^i$

2.5 Optimal-transport-based filters

Despite the advantages of the EnKF/ESRF, their performance in highly nonlinear environments is limited due to the intrinsic bias resulting from the low flexibility of their

linear updates. To improve the estimation precision, the filters based on optimal transport (OT) [53, 97] were proposed, achieving nonlinear transport between the prior and posterior ensembles. In this section, we first introduce the OT theory in Subsection 2.5.1, and then a review of the OT-based filters is provided in Subsection 2.5.2. Also, as a significant filter related to our proposed filter, the formulation of the SMF is explained in Subsection 2.5.3.

2.5.1 Optimal transport fundamentals

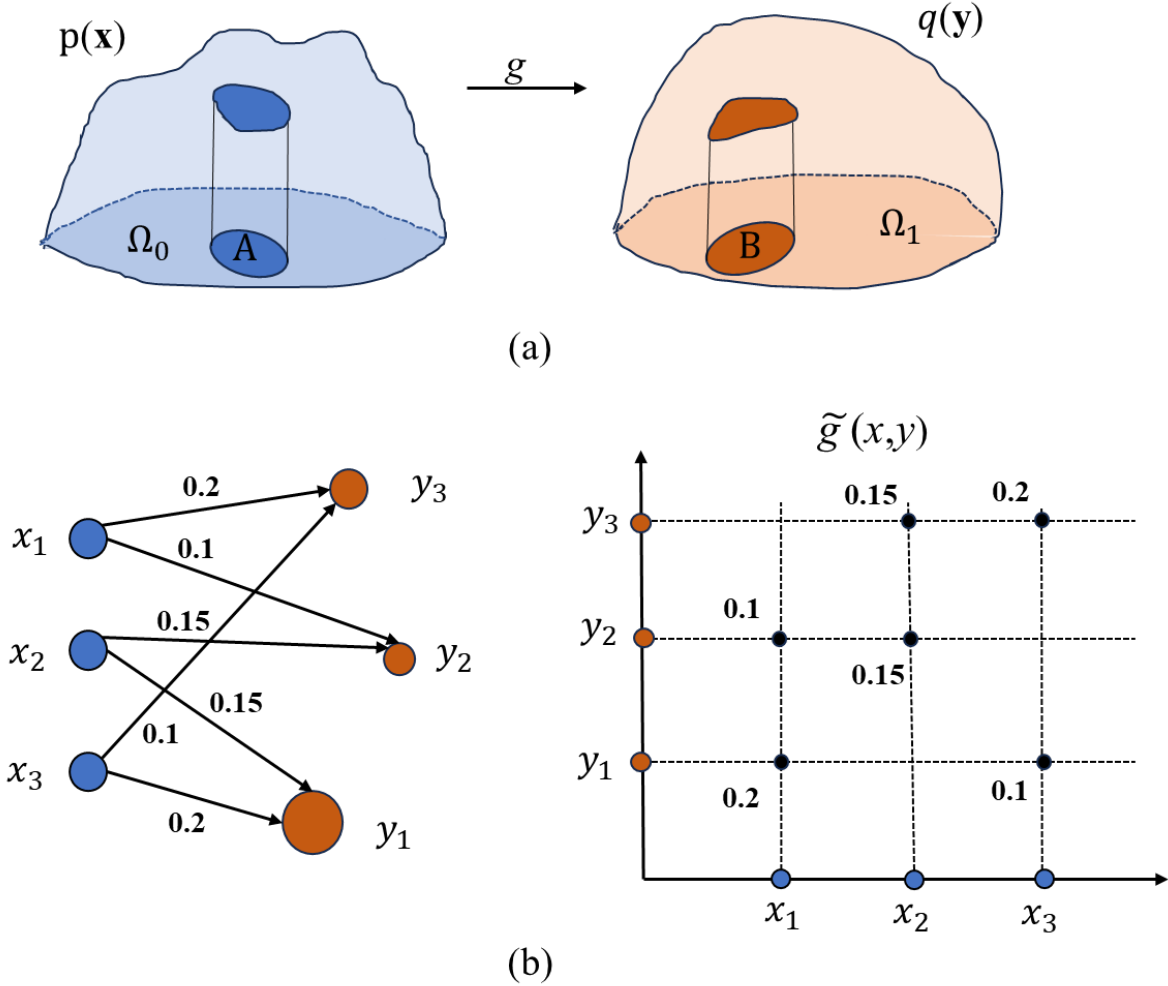


Figure 2.2: (a) Monge transport map (b) Kantorovich's transport plan [53].

The OT is a historical mathematical theory, which exploits the most efficient way to move one distribution to another. It was first presented by French mathematician

Gaspard Monge in 1781. Specifically, Monge's optimal transportation problem, as shown in Figure 2.2-(a), pursues a function g which pushes one probability distribution $p(\mathbf{x}), \mathbf{x} \in \Omega_0$ to another $q(\mathbf{y}), \mathbf{y} \in \Omega_1$ [53]. Also, an objective function needs to be optimised, i.e.,

$$(2.22) \quad J(p, q) = \inf_{g \in \text{MP}} \int_{\Omega_0} \text{cost}(\mathbf{x}, g(\mathbf{x})) p(\mathbf{x}) d\mathbf{x},$$

where the cost function $\text{cost}(\mathbf{x}, g(\mathbf{x}))$ denotes the transport cost from \mathbf{x} to $g(\mathbf{x})$ and $\text{cost}(\mathbf{x}, g(\mathbf{x})) = |\mathbf{x} - g(\mathbf{x})|$ at the original version. Besides, MP represents a measure preserving map meeting the condition

$$(2.23) \quad \int_{\mathbf{x}: g(\mathbf{x}) \in B} p(\mathbf{x}) d\mathbf{x} = \int_B q(\mathbf{y}) d\mathbf{y}$$

where $B \in \Omega_1$. If g is an injective function, we have

$$(2.24) \quad \int_A p(\mathbf{x}) d\mathbf{x} = \int_{g(A)} q(\mathbf{y}) d\mathbf{y}$$

where $A \in \Omega_0$ and g is called a transport map in this case. Further, if g is a smooth and one-to-one function, the differential form of (2.23) can be written as

$$(2.25) \quad \det(Dg(\mathbf{x})) q(g(\mathbf{x})) = p(\mathbf{x})$$

where Dg is the Jacobian of g . In general, there exists an infinite number of maps that solve equation (2.22). One way to select a map is to define a particular cost function and solve the OT problem. For example, g can be restricted by minimising the p-Wasserstein metric below

$$(2.26) \quad J(p, q) = \min_{g \in \text{MP}} \left(\int_{\Omega_0} \text{cost}(\mathbf{x}, g(\mathbf{x})) p(\mathbf{x}) d\mathbf{x} \right)^{\frac{1}{p}},$$

where $\text{cost}(\mathbf{x}, g(\mathbf{x})) = |\mathbf{x} - g(\mathbf{x})|^p$, $p \geq 1$.

In 1942, Monge's transport problem was generalised by Leonid V. Kantorovich, whose formulation pursues a transportation plan \tilde{g} and can be written as

$$(2.27) \quad J(p, q) = \min_{\tilde{g} \in \text{MP}} \int_{\Omega_0 \times \Omega_1} \text{cost}(\mathbf{x}, \mathbf{y}) d\tilde{g}(\mathbf{x}, \mathbf{y}),$$

where MP requires \tilde{g} to be a joint distribution of marginals $p(\mathbf{x})$ and $q(\mathbf{y})$, i.e.

$$(2.28) \quad \begin{aligned} \tilde{g}(\Omega_0 \times \mathbf{y}) &= q(\mathbf{y}) \\ \tilde{g}(\mathbf{x} \times \Omega_1) &= p(\mathbf{x}). \end{aligned}$$

Also, under the Wasserstein metric, the Kantorovich's formulation can be rewritten as

$$J(p, q) = \min_{\tilde{g} \in \text{MP}} \left(\int_{\Omega_0 \times \Omega_1} |\mathbf{x} - \mathbf{y}|^p d\tilde{g}(\mathbf{x}, \mathbf{y}) \right)^{\frac{1}{p}}.$$

Besides, we can represent the $p(\mathbf{x})$ and $q(\mathbf{y})$ as discrete distributions with M and N possible values, respectively, i.e.,

$$\begin{aligned} p(\mathbf{x}) &= \sum_{i=1}^M p_i \delta(\mathbf{x} - \mathbf{x}_i) \\ q(\mathbf{y}) &= \sum_{j=1}^N q_j \delta(\mathbf{y} - \mathbf{y}_j) \end{aligned},$$

where p_i and q_j represent the probabilities of \mathbf{x}_i and \mathbf{y}_j , respectively. Then the corresponding Kantorovich's problem can be written as

$$\begin{aligned} (2.29) \quad J(p, q) &= \min_{\tilde{g}} \sum_i \sum_j \text{cost}(\mathbf{x}_i, \mathbf{y}_j) \tilde{g}_{ij} \\ \text{s.t.} \quad \sum_j \tilde{g}_{ij} &= p_i, \quad \sum_i \tilde{g}_{ij} = q_j, \quad \tilde{g}_{ij} \geq 0 \end{aligned}$$

where \tilde{g}_{ij} represents the probability of $[\mathbf{x}_i^T, \mathbf{y}_j^T]^T$. In Figure 2.2-(b), an example of Kantorovich's transport plan based on the discrete distributions are presented.

We should note that the optimization problem over the joint distribution is over a different space than the MP used in equation (2.22). Specifically, for Monge's problem, a transport map assigns one element of \mathbf{x} to exactly one of \mathbf{y} . By contrast, Kantorovich's problem is more general and supports one-to-many movements.

2.5.2 OT-based filters

In recent years, the regularisation of many OT maps/plans has been presented and applied to filtering problems. [77] proposed the ensemble transform particle filter (ETPF), where the OT plan is designed based on linear programming. Although obtaining better filtering performance than the PF and EnKF, its computational load is $O(N^3 \log N)$. By contrast, [2] presented a differentiable particle filtering via Entropy-Regularized OT, of which the computational load is reduced to $O(N^2)$ owing to the efficient Sinkhorn Algorithm. Also, in [76], the mapping particle filter is introduced, where particles are transported from the prior to the posterior domain using a sequence of mappings which minimise the Kullback-Leibler (KL) divergence between the posterior and intermediate distributions. Nevertheless, its filtering efficiency is influenced by the slow convergence process. Besides, the stochastic and deterministic map filters are provided using the Knothe-Rosenblatt (KR) rearrangement in [87], and its computational load is $O(N)$.

2.5.3 Stochastic Map Filtering

In this section, the SMF framework is detailed. We first explain the KR rearrangement [97], which is a kind of OT map constituting the basis of the SMF approach. The cost function for the parameter optimisation of the KR rearrangement is introduced subsequently, and the data assimilation procedure of the SMF concludes this subsection.

2.5.3.1 Knothe-Rosenblatt rearrangement

Given any pair of positive densities, there exists a unique monotone triangular transport map, which defines a deterministic coupling between two distributions and is called the KR rearrangement. This strategy belongs to Monge's problem, i.e. a one-to-one transport map. Due to its ability to create connections between two probability densities, it has been widely used in Bayesian inference, importance sampling, SMC, etc [20, 65, 74].

A parametric strategy for the KR rearrangement is presented in [65]. Assume $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ and $\mathbf{y} = [y_1, y_2, \dots, y_n]^T$ are n -dimensional variables with distributions $p(\mathbf{x})$ and $q(\mathbf{y})$, respectively. Then, a standard triangular transport map can be expressed as

$$(2.30) \quad \mathbf{y} = S(\mathbf{x}) = \begin{bmatrix} S^1(x_1) \\ S^2(x_1, x_2) \\ \dots \\ S^n(x_1, x_2, \dots, x_n) \end{bmatrix}$$

where the map S transports the distribution $p(\mathbf{x})$ to $q(\mathbf{y})$ and each component of S is monotone with respect to its last input. The crucial property of the KR rearrangement for its application to the SMC algorithm is that it provides an implicit characterization of the marginal conditional distributions. In the example above, S^1 transports $p(x_1)$ to $q(y_1)$, S^2 transports $p(x_2|x_1)$ to $q(y_2|y_1)$ and so on. This conditional distribution transformation property is used in the design of the SMF algorithm.

The parameterisation of KR rearrangement is an important part of the design of transport maps. One way to parameterise each component of the map S is via multivariate polynomials which could either involve Hermite or Legendre polynomials [61]. [87] also provides a specific computational parameterisation of triangular maps, which is based on radial basis functions (RBFs).

2.5.3.2 Constructing KR rearrangement from samples

Following the definition of the triangular transport maps above, we now explain the cost function definition based on the KL divergence. We aim to optimise the parameters of the transport map polynomials which connect an arbitrary distribution $p(\mathbf{x})$ to a reference distribution $q(\mathbf{y})$. Following equation (2.22), since S is a monotone and differentiable transformation, we have

$$(2.31) \quad \hat{p}(\mathbf{x}) = q(S(\mathbf{x}))|\det(D_{\mathbf{x}}S)|,$$

where $\hat{p}(\mathbf{x})$ is an approximation of $p(\mathbf{x})$. The map S can be obtained by minimizing the difference between $\hat{p}(\mathbf{x})$ and $p(\mathbf{x})$, and the KL divergence can be used to measure this. It can be expressed as

$$(2.32) \quad \begin{aligned} D_{KL}(p|\hat{p}) &= \mathbb{E} \left[\log \left(\frac{p(\mathbf{x})}{\hat{p}(\mathbf{x})} \right) \right] \\ &= \mathbb{E}[\log(p(\mathbf{x})) - \log(q(S(\mathbf{x}))) - \log|\det(D_{\mathbf{x}}S)|]. \end{aligned}$$

Then, the transport map can be expressed in terms of the KL divergence as

$$(2.33) \quad \hat{S} = \arg \min_{S \in \mathcal{H}} \mathbb{E}[-\log(q(S(\mathbf{x}))) - \log|\det(D_{\mathbf{x}}S)|],$$

where \mathcal{H} is a function space for map S . The unknown term $\mathbb{E}[\log(p(\mathbf{x}))]$ is neglected in the objective function above because it is not related to S . We should note that in the above cost function, there is no term related to transport cost like equation (2.26) because the uniqueness of the transport map is guaranteed by the triangular structure and monotonicity of the KR rearrangement [65]. Assume that we have N samples $\mathbf{x}^i, i = 1 \dots N$, from $p(\mathbf{x})$, then the cost function based on the discrete samples is expressed as

$$(2.34) \quad \hat{S} = \arg \min_{S \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^N \left[-\log q(S(\mathbf{x}^i)) - \log|\det(D_{\mathbf{x}}S(\mathbf{x}^i))| \right].$$

Newton's method provides an efficient solution to the minimisation of the cost function in (2.34) and has been widely used in the literature [87].

2.5.3.3 Data Assimilation

In this section, we introduce the structure of the SMF. For a filtering problem, at any time k , we initially have samples $\mathbf{x}_{k-1|k-1}^i, i = 1, \dots, N$, from $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$. By sampling $p(\mathbf{x}_k|\mathbf{x}_{k-1|k-1}^i)$, we have $\mathbf{x}_{k|k-1}^i, i = 1, \dots, N$. Then by sampling $p(\mathbf{z}_k|\mathbf{x}_{k|k-1}^i)$, $\mathbf{z}_{k|k-1}^i, i =$

$1, \dots, N$ are obtained. Then we have $\left[\mathbf{x}_{k|k-1}^i, \mathbf{z}_{k|k-1}^i \right]^T, i = 1, \dots, N$ which are samples from the joint density $p(\mathbf{x}_k, \mathbf{z}_k | \mathbf{z}_{1:k-1})$. Following [87], we define the function $S^{\mathcal{X}}: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$, which is a part of the KR rearrangement $S: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n \times \mathbb{R}^m$, which transports the distribution $p(\mathbf{x}_k, \mathbf{z}_k | \mathbf{z}_{1:k-1})$ to a normal distribution. The resulting map is continuous and can be defined as

$$(2.35) \quad S^{\mathcal{X}}(\mathbf{z}_k, \mathbf{x}_k) \sim \mathcal{N}(0, \mathbf{I}_n)$$

From [87], we have that the analysis map T which transports the joint distribution to the posterior can be expressed as

$$(2.36) \quad T := S^{\mathcal{X}}(\mathbf{z}_k^*, \sim)^{-1} \circ S^{\mathcal{X}}(\mathbf{z}_k, \mathbf{x}_k)$$

where \mathbf{z}_k^* is a fixed observation at time k , $S^{\mathcal{X}}(\mathbf{z}_k^*, \sim)^{-1}$ denotes the inverse function of the map $S^{\mathcal{X}}(\mathbf{z}_k^*, \sim)$, and \circ denotes the composition of the two maps. In the above structure, $S^{\mathcal{X}}$ transforms the samples from the joint distribution to the standard normal which is then pushed by $S^{\mathcal{X}}(\mathbf{z}_k^*, \sim)^{-1}$ to the posterior. In principle, we can directly employ $S^{\mathcal{X}}(\mathbf{z}_k^*, \sim)^{-1}$ to produce posterior approximations by pushing forward samples from the standard normal density. However, from [87], transporting samples from $p(\mathbf{x}_k, \mathbf{z}_k | \mathbf{z}_{1:k-1})$ through T yields more accurate results which is attributed to the cancellation of errors in the composition of $\hat{S}^{\mathcal{X}}$ and its inverse.

In practice, we can obtain the estimator of T using the samples, and it can be expressed as

$$(2.37) \quad \hat{T} := \hat{S}^{\mathcal{X}}(\mathbf{z}_k^*, \sim)^{-1} \circ \hat{S}^{\mathcal{X}}(\mathbf{z}_k, \mathbf{x}_k)$$

where $\hat{S}^{\mathcal{X}}$ denotes the estimated map of $S^{\mathcal{X}}$.

In general, the process of stochastic map filtering can be divided into two steps, i.e. the forecast step and the analysis step. First, we get samples of $p(\mathbf{x}_k, \mathbf{z}_k | \mathbf{z}_{1:k-1})$ (forecasting). Second, the samples of the estimated posterior distribution can be obtained by transforming the joint prior samples through the map \hat{T} (analysis).

2.6 Hybrid filters

Except for the OT-based filters, an alternative strategy to improve the filter performance of the PF and the EnKF/ESRF is to build a hybrid filtering framework. Specifically, the likelihood function is first separated into two parts, and then the EnKF/ESRF and

PF-based filters are applied sequentially to assimilate each likelihood part, i.e.,

$$(2.38) \quad p(\mathbf{z}_k|\mathbf{x}_k) = p(\mathbf{z}_k|\mathbf{x}_k)^\varrho p(\mathbf{z}_k|\mathbf{x}_k)^{1-\varrho}$$

where $0 < \varrho < 1$ is the *splitting factor*, $p(\mathbf{z}_k|\mathbf{x}_k)^\varrho$ and $p(\mathbf{z}_k|\mathbf{x}_k)^{1-\varrho}$ are assimilated by different filters. As a result, the hybrid filters can yield more precise non-Gaussian approximations than PF since they alleviate the particle degeneracy issue through the use of EnKF/ESRF. Currently, three hybrid filters have been proposed and explored, namely the Gaussian mixture model-EnKF hybrid filter (GMM-EnKF) [26, 89]; the hybrid of ETPF and ESRF (ETPF-ESRF) [18, 77]; and the SIR-ESRFs [31] via combining the standard PF and ESRFs with a mean-preserving random orthogonal resampling [82]. Despite the improvement in the estimation precision, these filters also have some drawbacks. In GMM-EnKF, since the EnKF is implemented in the first stage, it is not suitable for moderately non-Gaussian models where prior distributions are non-Gaussian and posterior distributions are close to Gaussian. ETPF-ESRFs suffers from a large computational load due to the ETPF component. SIR-ESRFs provides an efficient solution to the moderately non-Gaussian models and the mean-preserving random orthogonal transformation in the ESRFs update stage produces the Gaussian approximation for the posterior distribution. However, when the posterior distribution is not close to a Gaussian form, this resampling technique causes large sampling errors. The details of GMM-EnKF and SIR-ESRFs are provided in Subsection 2.6.1 and 2.6.2, respectively, and their filtering performance is compared with our proposed PSMF in Chapter 3.

2.6.1 GMM-EnKF

For the GMM-EnKF, the $p(\mathbf{z}_k|\mathbf{x}_k)^\varrho$ and $p(\mathbf{z}_k|\mathbf{x}_k)^{1-\varrho}$ are assimilated by the stochastic EnKF and the PF, respectively. Assuming a Gaussian likelihood, according to the EnKF framework, the assimilation results of $p(\mathbf{z}_k|\mathbf{x}_k)^\varrho$ can be expressed as

$$\mathbf{x}_{k,i}^m = \mathbf{x}_{k|k-1}^i + \mathbf{K}_k^m \left(\mathbf{z}_k + \mathbf{v}_k^i - \mathbf{H}\mathbf{x}_{k|k-1}^i \right)$$

where $\mathbf{x}_{k,i}^m$ is the i -th assimilated sample by $p(\mathbf{z}_k|\mathbf{x}_k)^\varrho$ at time k , $\mathbf{v}_k^i \sim \mathcal{N}(\mathbf{0}, \frac{\mathbf{R}}{\varrho})$ and \mathbf{K}_k^m is the approximated Kalman gain calculated by

$$\begin{aligned} \mathbf{K}_k^m &= \mathbf{P}_k^{\mathbf{xx}} \mathbf{H}^T \left(\mathbf{H} \mathbf{P}_k^{\mathbf{xx}} \mathbf{H}^T + \mathbf{R}_\varrho \right)^{-1} \\ \mathbf{P}_k^{\mathbf{xx}} &= \frac{1}{N-1} \sum_{i=1}^N \left(\mathbf{x}_{k|k-1}^i - \bar{\mathbf{x}}_{k|k-1} \right) \left(\mathbf{x}_{k|k-1}^i - \bar{\mathbf{x}}_{k|k-1} \right)^T \\ \mathbf{R}_\varrho &= \frac{\mathbf{R}}{\varrho} \end{aligned}$$

Then the assimilation posterior can be written as a Gaussian Mixture distribution, i.e.,

$$\frac{1}{N} \sum_{i=1}^N \mathcal{N}(\bar{\mathbf{x}}_{k,i}^m, \mathbf{P}_k^m),$$

where

$$\begin{aligned} \bar{\mathbf{x}}_{k,i}^m &= \mathbf{x}_{k|k-1}^i + \mathbf{K}_k^m (\mathbf{z}_k - \mathbf{H} \mathbf{x}_{k|k-1}^i) \\ \mathbf{P}_k^m &= \frac{1}{\rho} \mathbf{K}_k^m \mathbf{R} \mathbf{K}_k^{mT} \end{aligned}$$

Instead of performing the standard EnKF sampling from this density, we delay that sampling and perform the multiplication with the second likelihood $p(\mathbf{z}_k | \mathbf{x}_k)^{1-\rho}$ analytically. The full posterior can be represented as a Gaussian mixture density, i.e.,

$$\sum_{i=1}^N w_i \mathcal{N}(\bar{\mathbf{x}}_{k|k}^i, \mathbf{P}_{k|k}),$$

where

$$\begin{aligned} \bar{\mathbf{x}}_{k|k}^i &= \bar{\mathbf{x}}_{k,i}^m + \mathbf{K}_{k|k} (\mathbf{z}_k - \mathbf{H} \bar{\mathbf{x}}_{k,i}^m) \\ \mathbf{P}_{k|k} &= (\mathbf{I} - \mathbf{K}_{k|k} \mathbf{H}) \mathbf{P}_k^m \\ w_i &\propto \mathcal{N} \left(\mathbf{z}_k - \mathbf{H} \bar{\mathbf{x}}_{k,i}^m, \mathbf{H} \mathbf{P}_k^m \mathbf{H}^T + \frac{\mathbf{R}}{(1-\rho)} \right) \\ \mathbf{K}_{k|k} &= \mathbf{P}_k^m \mathbf{H}^T \left(\mathbf{H} \mathbf{P}_k^m \mathbf{H}^T + \frac{\mathbf{R}}{(1-\rho)} \right)^{-1} \end{aligned}$$

Then the posterior samples $\mathbf{x}_{k|k}^i$ can be obtained by sampling from the Gaussian mixture distribution. The procedure of the GMM-EnKF is detailed in Algorithm 8.

Algorithm 8: GMM-EnKF framework

Input: $\mathbf{x}_{k-1|k-1}^i$

Prediction:

$$\mathbf{x}_{k|k-1}^i = f(\mathbf{x}_{k-1|k-1}^i) + \mathbf{w}_{k-1}^i$$

Assimilate $p(\mathbf{z}_k | \mathbf{x}_k)^\rho$ and obtain:

$$\frac{1}{N} \sum_{i=1}^N \mathcal{N}(\bar{\mathbf{x}}_{k,i}^m, \mathbf{P}_k^m)$$

Assimilate $p(\mathbf{z}_k | \mathbf{x}_k)^{1-\rho}$ and obtain:

$$\sum_{i=1}^N w_i \mathcal{N}(\bar{\mathbf{x}}_{k|k}^i, \mathbf{P}_{k|k});$$

Obtain $\mathbf{x}_{k|k}^i$ by sampling from the posterior Gaussian mixture distribution;

return $\mathbf{x}_{k|k}^i$

2.6.2 SIR-ESRF

For the SIR-ESRFs, the $p(\mathbf{z}_k|\mathbf{x}_k)^\ell$ is assimilated by the PF in the first step. Sampling the prior particles $\mathbf{x}_{k|k-1}^i$, the corresponding sample weights can be calculated by

$$(2.39) \quad w_k^i \propto p(\mathbf{z}_k|\mathbf{x}_{k|k-1}^i)^\ell.$$

Then obtain $\mathbf{x}_{k,i}^m$ by resampling the discrete distribution. In the second step, a mean-preserving random orthogonal transformation [82] is adopted to increase the diversity of the particles. Hence, compared with the ESRFs in Subsection 2.4.2, the analysis ensemble perturbation matrix \mathbf{A}^a is replaced with

$$\overrightarrow{\mathbf{A}}^a = \mathbf{A}^a \mathbf{U} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{P} \end{bmatrix} \mathbf{U}^T$$

The matrix \mathbf{U} is an orthogonal matrix whose first column is proportional to $\mathbf{1}$, while the matrix \mathbf{P} is a random orthogonal matrix of size $N-1 \times N-1$. Then the posterior ensemble can be constituted by

$$\mathbf{x}_{k|k}^i = \overline{\mathbf{x}}_{k|k} + \sqrt{N-1} \overrightarrow{\mathbf{A}}^a_i$$

where $\overrightarrow{\mathbf{A}}^a_i$ is i -th column of $\overrightarrow{\mathbf{A}}^a$.

Algorithm 9: SIR-ESRF framework

Input: $\mathbf{x}_{k-1|k-1}^i$

Prediction:

$$\mathbf{x}_{k|k-1}^i = f(\mathbf{x}_{k-1|k-1}^i) + \mathbf{w}_{k-1}^i$$

Assimilate $p(\mathbf{z}_k|\mathbf{x}_k)^\ell$ based on PF, and then obtain $\mathbf{x}_{k,i}^m$;

Assimilate $p(\mathbf{z}_k|\mathbf{x}_k)^{1-\ell}$ based on ESRFs, and then obtain $\mathbf{x}_{k|k}^i$;

return $\mathbf{x}_{k|k}^i$

2.7 Data-driven filters

In the previous sections, we introduced various traditional model-based (MB) filters, which rely on specific assumptions about the system dynamics and measurement models. If these assumptions are violated or inaccurate, the filtering performance may degrade significantly. With the development of deep neural networks (DNNs), DD filters [28] were proposed, where accurate knowledge of the state-space model is not required.

Nonetheless, these DD approaches lack interpretability and require many large data sets even for simple sequences. Following this, a hybrid filter, KalmanNet [79], combining DD with MB filters was presented. The hybrid framework is more interpretable due to its structure based on MB filters and can produce precise estimates owing to the trainable neural network. In Subsection 2.7.1, we first introduce the deep neural network. Then, as a significant structure for the DD filter, the recurrent neural network (RNN) [83] is explained in Subsection 2.7.2. Further, the DD and DD-MB filters are explained in Subsection 2.7.3 and Subsection 2.7.4, respectively.

2.7.1 DNN fundamentals

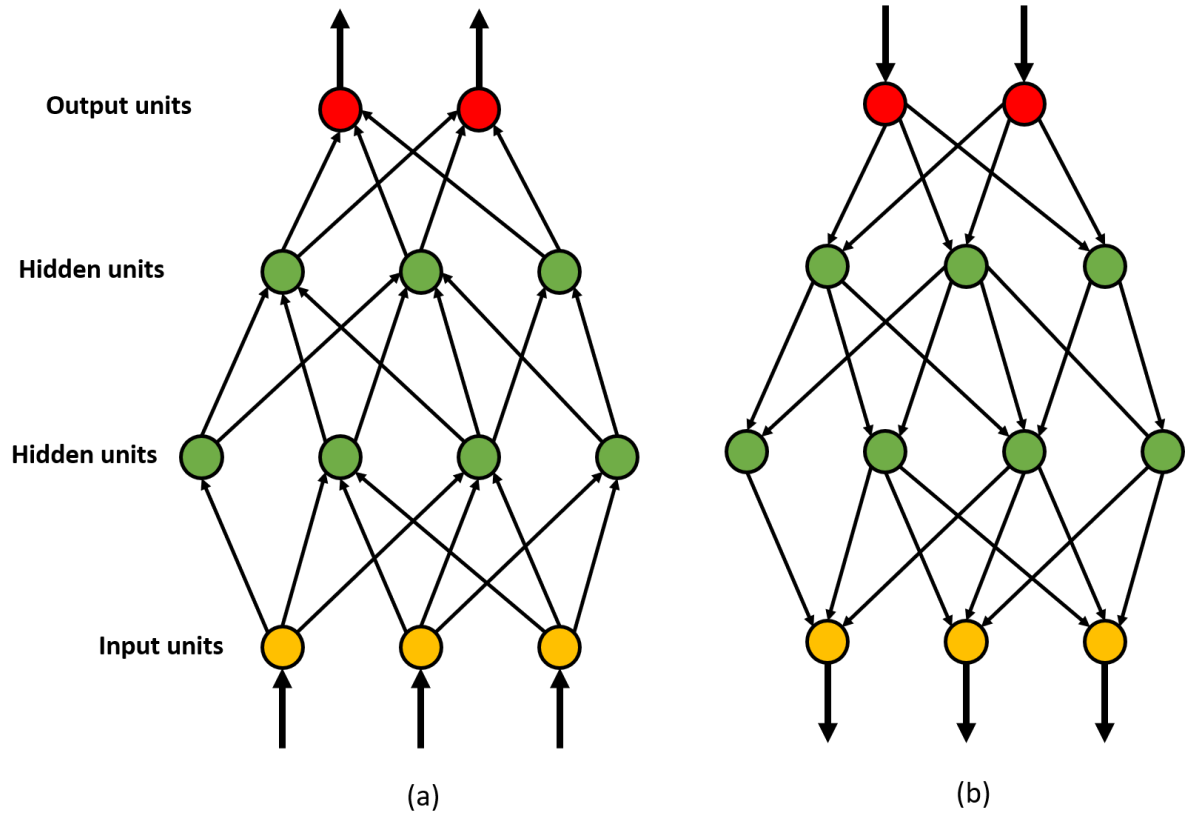


Figure 2.3: Deep neural network structure. (a) and (b) plot the forward pass and back-propagation steps, respectively.

Neural networks [103] are a class of machine learning models inspired by the structure of the human brain. They consist of interconnected nodes, called neurons, organized into layers. Each neuron takes inputs, performs computation and produces an output.

DNNs [57] are neural networks with multiple hidden layers between the input and output layers. These hidden layers enable the network to learn hierarchical representations of the input data, capturing complex patterns and relationships. To illustrate the structure, an exemplary multilayer neural network is shown in Figure 2.3, which follows a feedforward architecture. The information flows from the input layer through the hidden layers to the output layer. Each layer contains multiple neurons that apply a nonlinear activation function to the weighted sum of their inputs.

In the training process, there are two steps: forward pass and backpropagation. During the forward pass, the input data is fed into the network, and the outputs of each layer are computed sequentially as shown in Figure 2.3-(a). By contrast, Figure 2.3-(b) shows the backpropagation step, where the gradients of the model parameters with respect to the loss function are calculated by chain rule and gradient-based optimization algorithms are employed to update the parameters.

2.7.2 RNN

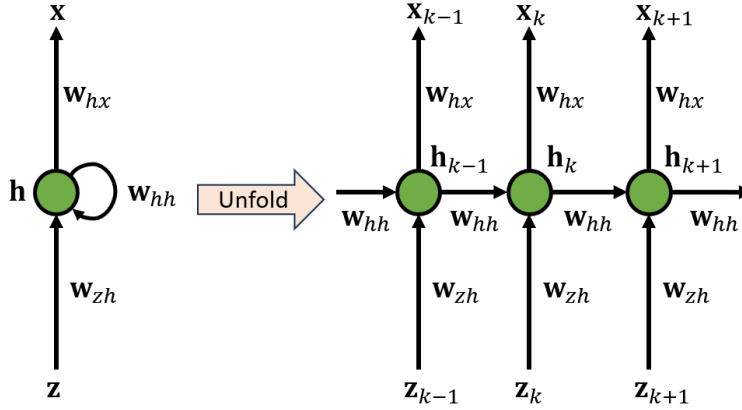


Figure 2.4: RNN

Following the introduction of the DNN in the last section, this section focuses on the RNN, which is a type of artificial neural network dealing with sequential data or time series data. As shown in Figure 2.4, at any time k , the RNN processes the input \mathbf{z}_k and maintains the hidden states \mathbf{h}_{k-1} , containing past information. Hence, the recurrent nature allows the network to capture temporal dependencies in the sequential data. In this subsection, three typical RNN structures are explained, including vanilla RNN [16], long short-term memory (LSTM) network [28] and gated recurrent unit (GRU) [16].

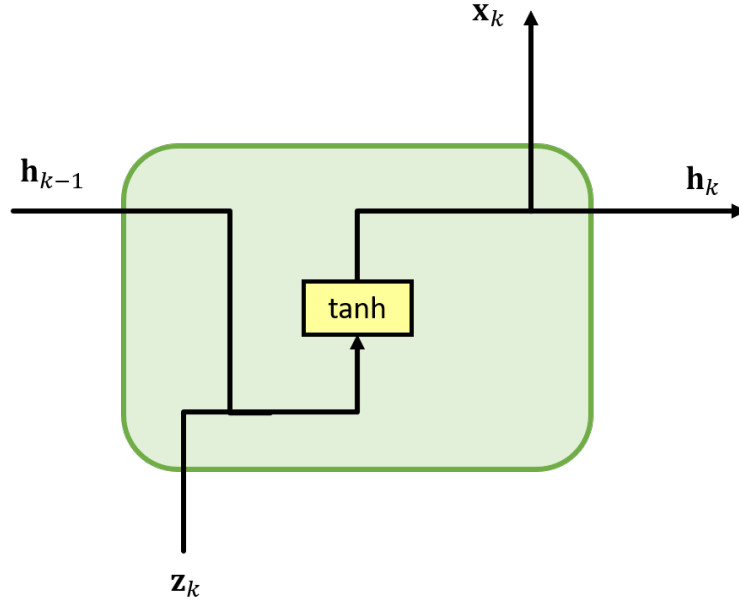


Figure 2.5: vanilla RNN

Vanilla RNN is the most simple form of RNN and maintains only one hidden state as shown in Figure 2.5. The computation in a vanilla RNN can be described by the following equations:

$$\mathbf{h}_k = \tanh(\mathbf{W}_{hh}\mathbf{h}_{k-1} + \mathbf{W}_{zh}\mathbf{z}_{k-1})$$

The prediction of \mathbf{x}_k can be obtained by a matrix projection on top of the hidden state, i.e.,

$$\mathbf{x}_k = \mathbf{W}_{xh}\mathbf{h}_k$$

The parameters (weights) in a vanilla RNN are shared across all time steps, allowing the network to capture the temporal patterns in the sequential data. However, its performance is limited in learning long-term dependencies due to the vanishing gradient problem. Specifically, the network is trained using backpropagation through time, and then the gradients can diminish or explode exponentially when they propagate through multiple time steps.

To alleviate the vanishing gradient problem, the LSTM structure introduces a memory cell and three gating mechanisms: input gate, forget gate, and output gate as shown in Figure 2.6. These gates regulate the information flow within the LSTM cell, allowing it to retain important information over long sequences and forget irrelevant information.

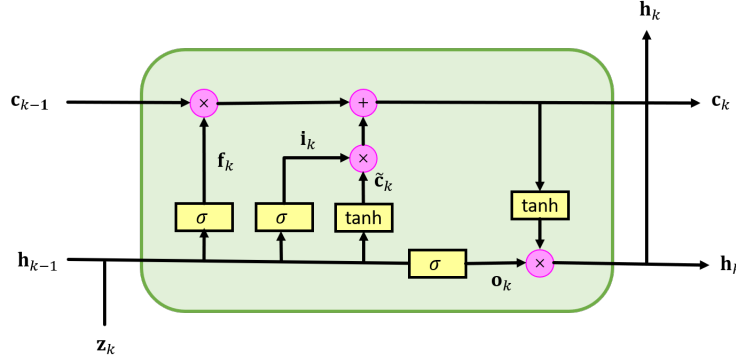


Figure 2.6: LSTM

The formulation of the LSTM can be written as

$$\mathbf{f}_k = \sigma(\mathbf{W}_{hf}\mathbf{h}_{k-1} + \mathbf{W}_{zf}\mathbf{z}_k)$$

$$\mathbf{i}_k = \sigma(\mathbf{W}_{hi}\mathbf{h}_{k-1} + \mathbf{W}_{zi}\mathbf{z}_k)$$

$$\mathbf{o}_k = \sigma(\mathbf{W}_{ho}\mathbf{h}_{k-1} + \mathbf{W}_{zo}\mathbf{z}_k)$$

$$\tilde{\mathbf{c}}_k = \tanh(\mathbf{W}_{hg}\mathbf{h}_{k-1} + \mathbf{W}_{zg}\mathbf{z}_k)$$

$$\mathbf{c}_k = \mathbf{f}_k \circ \mathbf{c}_{k-1} + \mathbf{i}_k \circ \tilde{\mathbf{c}}_k$$

$$\mathbf{h}_k = \mathbf{o}_k \circ \tanh(\mathbf{c}_k)$$

where the cell state \mathbf{c}_k is the main component, which carries information over time and is updated by combining the previous cell state \mathbf{c}_{k-1} with the new candidate cell state $\tilde{\mathbf{c}}_k$. Also, the forget gate vector \mathbf{f}_k takes the previous hidden state \mathbf{h}_{k-1} and the current input \mathbf{z}_k as inputs, and determines how much of the previous cell state \mathbf{c}_{k-1} to retain or forget. Further, the input gate vector \mathbf{i}_k determines how much new information should be added to the cell state, and the output gate vector \mathbf{o}_k determines the amount of information to pass to the next hidden state \mathbf{h}_k .

To simplify the LSTM structure, the GRU structure was proposed, combining the forget and input gates into a single update gate and merging the cell state and hidden state. The GRU has fewer parameters than LSTM and hence is more computationally efficient. The GRU structure is illuminated in Figure 2.7, and the corresponding formulation can be written as

$$\mathbf{y}_k = \sigma(\mathbf{W}_{hz}\mathbf{h}_{k-1} + \mathbf{W}_{yz}\mathbf{z}_k)$$

$$\mathbf{r}_k = \sigma(\mathbf{W}_{hr}\mathbf{h}_{k-1} + \mathbf{W}_{zr}\mathbf{z}_k)$$

$$\tilde{\mathbf{h}}_k = \tanh(\mathbf{W}_{hn}(\mathbf{r}_k \circ \mathbf{h}_{k-1}) + \mathbf{W}_{zn}\mathbf{z}_k)$$

$$\mathbf{h}_k = (1 - \mathbf{y}_k) \circ \mathbf{h}_{k-1} + \mathbf{y}_k \circ \tilde{\mathbf{h}}_k$$

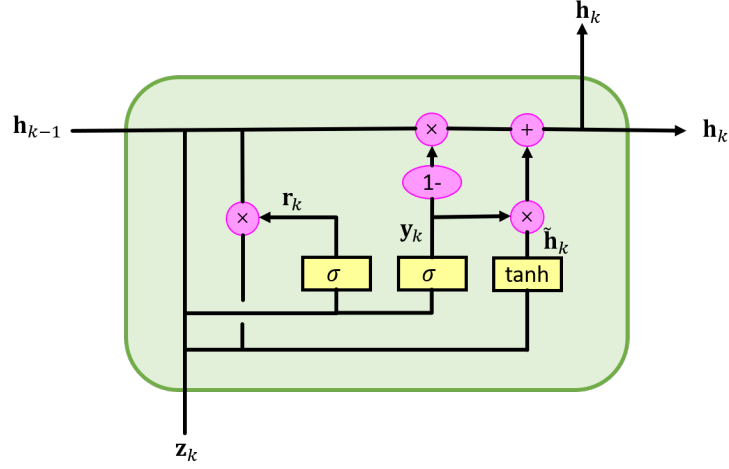


Figure 2.7: GRU

where the candidate hidden state $\tilde{\mathbf{h}}_k$ is a temporary memory that holds the information to be considered for the current hidden state. Also, the update gate vector \mathbf{y}_k determines how much of the previous hidden state \mathbf{h}_{k-1} should be passed to the current time step. Further, the reset gate vector \mathbf{r}_k controls the degree to which \mathbf{h}_{k-1} is taken into account when computing the current hidden state \mathbf{h}_k .

2.7.3 DD filters

Based on the introduced RNN structures, there are three ways to achieve DD filters. First, the recurrent neural networks take in the observations and sequentially output the true states [28] as shown in Figure 2.8-(a). However, to cope with the complexity of uncertain real-world data and achieve better belief approximation, a long latent vector \mathbf{h}_k is needed, which thus increases the number of network parameters and the amount of data required for training. The second approach, PF-RNN, combines the strengths of RNNs and particle filtering by learning latent particles $\mathbf{h}_k^i, i = 1 \dots N$ and updating the belief with a particle filter [60]. Without lengthening the latent vector \mathbf{h}_k , the required data amount is reduced. By contrast, the third method, LSTM-KF [19], proposes to learn the parameters of the KF by LSTM as shown in Figure 2.9.

2.7.4 DD-MB filtering methods

To reduce the required data amount, [79] proposed a hybrid DD-MB filter, call Kalman-Net, combining the MB and DD methods as shown in Figure 2.10. Specifically, at any

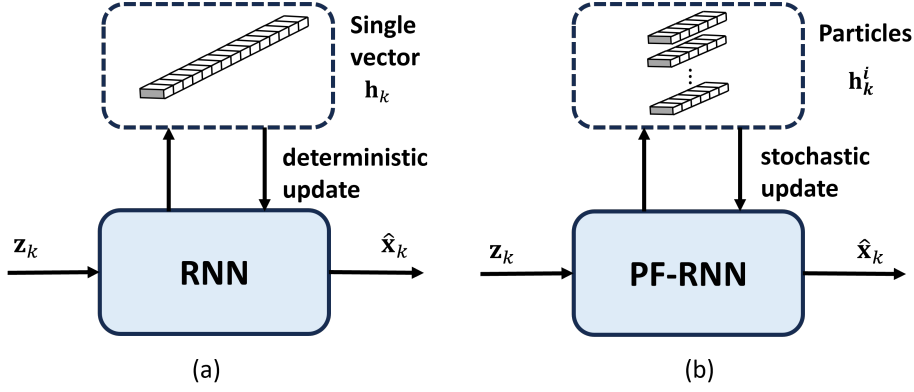


Figure 2.8: (a) and (b) show the original RNN-based filter and the PF-RNN-based filter, respectively.

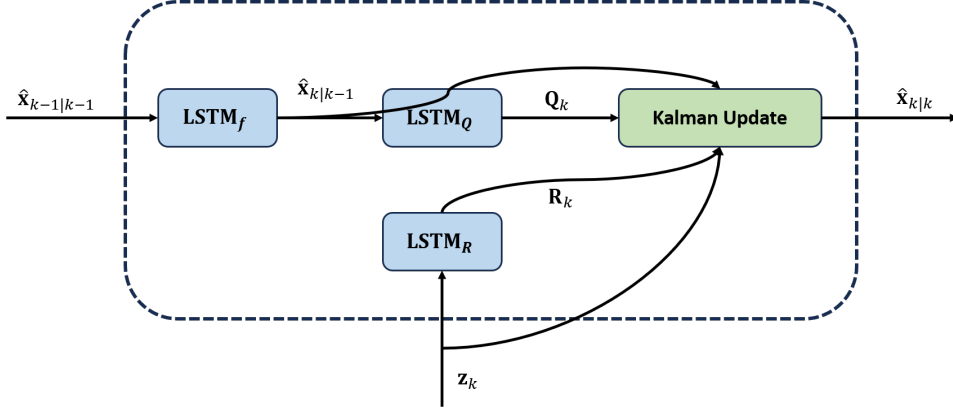


Figure 2.9: The LSTM-KF-based filter.

time k , the posterior estimation requires two steps: prediction and update. During the prediction step, the prior estimates for the current state $\hat{\mathbf{x}}_{t|t-1}$ and observation $\hat{\mathbf{y}}_{t|t-1}$ are computed based on the known transition function $f()$ and measurement function $h()$. Then, the update step assimilates the new observation \mathbf{y}_t based on the outputted Kalman gain of a dedicated RNN structure. Also, the input features of the RNN are selected as the innovation difference $\Delta \mathbf{y}_t = \mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1}$ and forward update difference $\Delta \hat{\mathbf{x}}_{t-1} = \hat{\mathbf{x}}_{t-1|t-1} - \hat{\mathbf{x}}_{t-1|t-2}$. Additionally, [79] presents an efficient supervised training scheme, of which the corresponding Loss function is based on the estimation error $\mathbf{x}_t - \hat{\mathbf{x}}_{t|t}$. Although the KalmanNet performs better than the traditional filters, it requires the ground-truth hidden state sequences, which may be costly to obtain. Then, an unsupervised KalmanNet [78] was proposed, and its Loss function is based on the observation prediction error $\mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1}$.

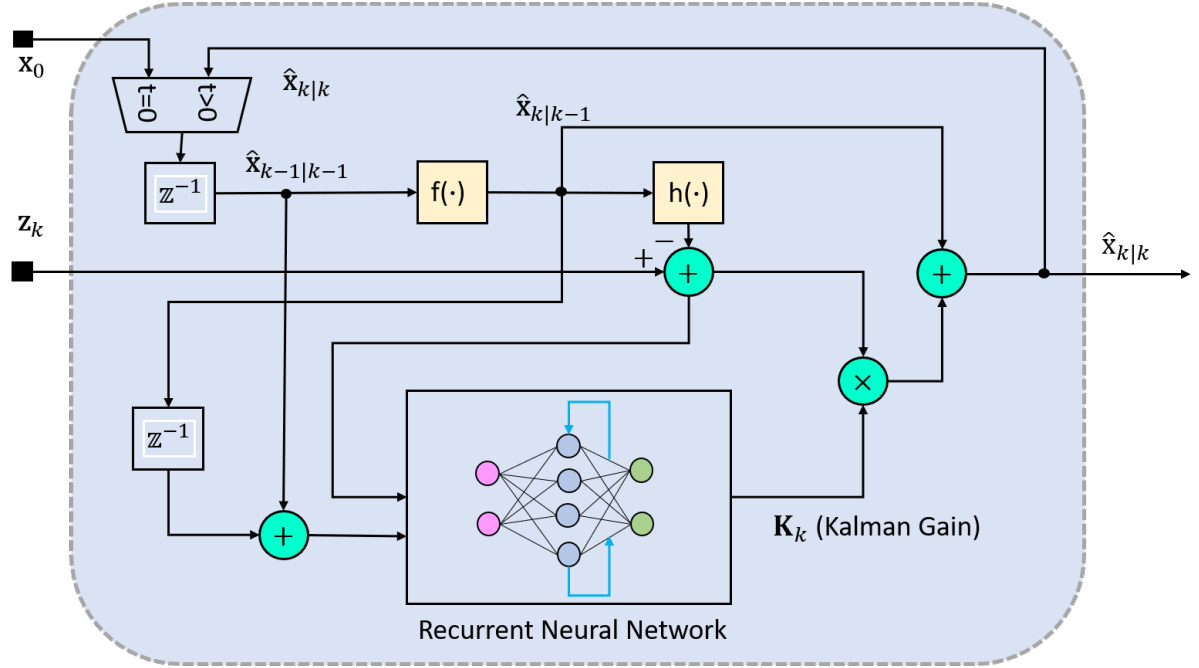


Figure 2.10: KalmanNet

Compared with traditional DD methods, the KalmanNet methods are based on the EKF framework. Hence, it is more interpretable, infers a reduced complexity and can be trained with a relatively small data set. Also, due to the introduction of the dedicated RNN, the KalmanNet can accurately characterize the state dynamics and obtain more precise estimates than the MB filters, where the estimation precision is influenced by the model error.

2.8 Conclusion

This chapter provides a comprehensive review of the previous work related to our proposed filters. We start from the classic KF and its nonlinear extensions, namely the EKF and UKF. Also, Section 2.2 explains the RKF framework for the linear models with heavy-tailed noise. Although both the KFs and RKFs are efficient, their performance is limited under highly-nonlinear models. As the most important filter for the high-nonlinear dynamics, the PFs are detailed in Section 2.3, and then the resampling techniques and various importance proposals for the alleviation of the particle degeneracy are also provided. By contrast, Section 2.4 introduces the EnKF/ESRFs framework, where the sample weights are always equal. However, its filtering performance is limited due

to the intrinsic bias resulting from the linear transformation. By contrast, the OT-based filters explained in Section 2.5 achieve nonlinear transport between the prior and posterior domains. To utilise the advantages of the PF and the transport-based filters, including EnKF/ESRFs and OT-based methods, several hybrid filters were proposed, where the likelihood function is separated by a splitting factor as explained in Section 2.6. For all these particle-based filters, the computational load is expensive due to their reliance on numerous samples. Except for these MB filters, Section 2.7 introduces several DD filters, which provide novel solutions for the filtering tasks based on the deep-learning technique.

A HYBRID PARTICLE-STOCHASTIC MAP FILTER

Filtering in nonlinear non-Gaussian state-space models is known to be a challenging task due to the posterior distribution being either intractable or expressed in a complex form. Following the review of various existing filters provided in the last chapter, this chapter proposes a novel nonlinear filter, PSMF, which combines the PF and SMF. As explained, although generally outperforming the traditional filters, the PF suffers from sample degeneracy. By contrast, drawing from optimal transport theory, the SMF [87] accommodates a solution to this problem. However, its performance is influenced by the limited flexibility of nonlinear map parameterisation. To alleviate the drawbacks of the PF and SMF, the PSMF incorporates these two filters into one filtering framework.

The rest of this chapter is organised as follows: In Section 3.1, we present the details of the proposed PSMF algorithm. The experimental analysis is performed in Section 3.2, whilst Section 3.3 summarises the chapter.

3.1 Stochastic Map-Augmented Particle Filtering

In this section, we present the proposed hybrid filter that combines PF with the SMF based on the nonlinear state space model in equation (1.2). At any time k , we have samples $\mathbf{x}_{k-1}^i \sim p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}^*)$, $i = 1, \dots, N$. Then, forecast samples $(\mathbf{x}_k^i, i = 1, \dots, N)$ from $p(\mathbf{x}_k|\mathbf{z}_{1:k-1}^*)$ can be obtained by sampling from $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$. After the prior samples are

obtained, the likelihood function is separated as

$$(3.1) \quad p(\mathbf{z}_k^* | \mathbf{x}_k) = p(\mathbf{z}_k^* | \mathbf{x}_k)^\rho p(\mathbf{z}_k^* | \mathbf{x}_k)^{1-\rho}$$

where $0 < \rho < 1$ is the *splitting factor*, $p(\mathbf{z}_k^* | \mathbf{x}_k)^\rho$ is assimilated by PF and $p(\mathbf{z}_k^* | \mathbf{x}_k)^{1-\rho}$ is assimilated by SMF. For ρ values of 0 and 1, the hybrid filter boils down to the SMF and the PF, respectively.

When ρ is small, PSMF is close to the SMF. In this case, the sample degeneracy can be alleviated effectively, but the estimation error increases due to the limited flexibility of the maps. We should note that theoretically, the degree of non-linearity of a map can be arbitrarily high and there is no limit to its flexibility. However, highly non-linear maps suffer from larger map variances and may produce worse estimations than linear maps. So, the non-linearity of maps should be controlled according to ensemble sizes. Specially, with a few samples, linear or low-order non-linear maps should be used. By contrast, a larger number of particles can restrict the variance of a highly non-linear map [87].

For large ρ , PSMF approaches closely the PF. Then, the error from sample degeneracy increases. In this case, as $1 - \rho$ tends to zero, $p(\mathbf{z}_k^* | \mathbf{x}_k)^{1-\rho}$ just reshapes the distribution slightly. So, there is no need for a highly flexible transport map to complete the second assimilation step, and variances of low-order maps are small. With the adaptability mentioned above, the proposed hybrid filter achieves a trade-off between the PF and the SMF.

Before we implement the assimilation step, ρ needs to be chosen. For a given $0 \leq \theta \leq 1$, we choose ρ as the largest value which satisfies the condition

$$(3.2) \quad ESS \geq \theta N$$

where ESS is the effective sample size [26] and can be approximated by the sample weights. For a given ρ , based on (2.15), we have the weight update equation which can be expressed as

$$(3.3) \quad w_k^i \propto w_{k-1}^i p(\mathbf{z}_k^* | \mathbf{x}_k^i)^\rho.$$

Since a resampling step is taken at every iteration, we have $w_{k-1}^i = \frac{1}{N}$ and the normalized particle weights can be written as

$$(3.4) \quad w_k^i = \frac{p(\mathbf{z}_k^* | \mathbf{x}_k^i)^\rho}{\sum_{j=1}^N p(\mathbf{z}_k^* | \mathbf{x}_k^j)^\rho}.$$

Then, the ESS can be approximated by

$$(3.5) \quad ESS = \frac{1}{\sum_i (w_k^i)^2}$$

From [26], searching the value of $0 \leq \rho \leq 1$ satisfying the condition in (3.2) can be achieved by a root finding method. For large θ , the searched ρ is small, and then the hybrid filter is closer to SMF. For a smaller θ , the hybrid filter moves towards the PF. The optimal values of θ depend on the nonlinear models. Currently, there is no automatic way to select parameter θ , and it needs to be selected manually based on the models. After θ is selected and ρ is calculated, the PF is implemented to update $p(\mathbf{z}_k^*|\mathbf{x}_k)^\rho$. Based on (3.4), we obtain the normalized particle weights corresponding to the selected ρ .

To alleviate the particle degeneracy problem, we utilise a systematic resampling, followed by a smoothing step. The details of the resampling method are explained in section 2.3.3. Different strategies for the alleviation of particle degeneracy [18, 26, 31] have been proposed in the literature. For example, GMM-EnKF avoids the degeneracy by sampling from the posterior density approximated by a Gaussian Mixture (GM); the ETPF-ESRF filter employs the optimal transport strategy to implement the resampling; a mean-preserving random orthogonal transformation is used in SIR-ESRF to break the degeneracy. Although the performance of systematic resampling is proved in previous work, this method has not been embedded in hybrid filters, because the discontinuity update influences its application in the high-dimensional scenario. In this work, we only consider low-dimension cases, but with the developments of the local PF [75, 96] and its combination with the EnKF [80], we can reasonably consider applying our hybrid filter to high-dimension models. This however is out of the scope of this chapter.

After resampling, all the particle weights become equal, $w_k^i = 1/N$. To increase the diversity of particles, we add a small Gaussian noise to smooth the samples, which is helpful to decrease the variance of transport maps in the SMF assimilation stage.

$$(3.6) \quad \mathbf{x}_k^i = \bar{\mathbf{x}}_k + \zeta(\mathbf{x}_k^i - \bar{\mathbf{x}}_k) + \beta_{\text{inf}} * \tilde{\mathbf{v}}_k^i, \tilde{\mathbf{v}}_k^i \sim N(0, \text{var}(\mathbf{x}_k))$$

where $0 < \beta_{\text{inf}} < 1$ is the smoothing factor and $\text{var}(\mathbf{x}_k)$ is the variance of the resampled samples. $\zeta = \sqrt{1 - \beta_{\text{inf}}^2}$ is the shrinking factor, which is employed to remove the excess variance caused by the added noise [89].

Next, $p(\mathbf{z}_k^*|\mathbf{x}_k)^{1-\rho}$ is assimilated by the SMF. First, to obtain the samples $[\mathbf{z}_k^i, \mathbf{x}_k^i]$ from the joint distribution, we sample the likelihood $p(\mathbf{z}_k|\mathbf{x}_k^i)^{1-\rho}$, $i = 1, \dots, N$. In this work, we only consider Gaussian and GM likelihood. For Gaussian cases, assume $p(\mathbf{z}_k|\mathbf{x}_k^i) = \mathcal{N}(f(\mathbf{x}_k^i), \Sigma)$, i.e.

$$p(\mathbf{z}_k|\mathbf{x}_k^i) = \frac{1}{(2\pi)^{\frac{m}{2}} |\Sigma|^{\frac{1}{2}}} \exp \left[-\frac{1}{2} \left(\mathbf{z}_k - f(\mathbf{x}_k^i) \right)^T \Sigma^{-1} \left(\mathbf{z}_k - f(\mathbf{x}_k^i) \right) \right]$$

where Σ is the covariance matrix. Then

$$\begin{aligned} p(\mathbf{z}_k | \mathbf{x}_k^i)^{1-\rho} &= \left[\frac{1}{(2\pi)^{\frac{m}{2}} |\Sigma|^{\frac{1}{2}}} \right]^{1-\rho} \exp \left[-\frac{1}{2} \left(\mathbf{z}_k - f(\mathbf{x}_k^i) \right)^T \left(\frac{\Sigma}{1-\rho} \right)^{-1} \left(\mathbf{z}_k - f(\mathbf{x}_k^i) \right) \right] \\ &\propto \frac{1}{(2\pi)^{\frac{m}{2}} \left| \frac{\Sigma}{1-\rho} \right|^{\frac{1}{2}}} \exp \left[-\frac{1}{2} \left(\mathbf{z}_k - f(\mathbf{x}_k^i) \right)^T \left(\frac{\Sigma}{1-\rho} \right)^{-1} \left(\mathbf{z}_k - f(\mathbf{x}_k^i) \right) \right] \\ &= \mathcal{N}(f(\mathbf{x}_k^i), \frac{\Sigma}{1-\rho}). \end{aligned}$$

Hence, we can sample $\mathcal{N}(f(\mathbf{x}_k^i), \frac{\Sigma}{1-\rho})$ to obtain \mathbf{z}_k^i . For GM cases, the likelihood can be expressed as $p(\mathbf{z}_k | \mathbf{x}_k^i) = \sum_{j=1}^M p_j \mathcal{N}(f(\mathbf{x}_k^i), \Sigma_j)$ where M is the number of mixtures. p_j and Σ_j are the mixing factor and the covariance matrix of the i -th Gaussian component, respectively. Similarly, we can sample $\sum_{j=1}^M p_j \mathcal{N}(f(\mathbf{x}_k^i), \frac{\Sigma_j}{1-\rho})$ to obtain \mathbf{z}_k^i .

Then, the map $S^{\hat{\mathcal{X}}}$ can be designed from the samples $[\mathbf{z}_k^i, \mathbf{x}_k^i]$ according to (2.35) and the transport map \hat{T} can be designed by (2.37).

In the last step, the samples $[\mathbf{z}_k^i, \mathbf{x}_k^i]$ are transported with the transport map \hat{T} as

$$(3.7) \quad \mathbf{x}_k^i = \hat{T}(\mathbf{z}_k^i, \mathbf{x}_k^i)$$

In our work, thanks to the generalisation of the EnKF, the transport map \hat{T} can be designed to be both linear and nonlinear which can then respectively be called PSMF-L and PSMF-NL. As also noted in [87], the nonlinear map achieves better performance compared to the linear map but suffers from larger map variances. The overall procedure of the proposed PSMF algorithm is summarised in Algorithm 10.

3.2 Simulation Results

3.2.1 State-Space Models

In this section, we introduce the low-dimensional nonlinear state-space models used in the experimental analysis. There are four benchmark models considered, i.e. (1) the univariate non-stationary growth model [62], (2) Henon map [31], (3) Lorenz-63 model [87], and (4) a target tracking model also used in [35, 36]. The first model is one-dimensional, the Henon map is a two-dimensional model, whilst Lorenz-63 and the target tracking models are of dimensions 3 and 4, respectively. The details of these state-space models are presented in the sequel.

Algorithm 10: The PSMF Algorithm

```

procedure PSMF ( $\mathbf{z}_k^*, p(\mathbf{x}_k|\mathbf{x}_{k-1}), p(\mathbf{z}_k|\mathbf{x}_k), \theta, \mathbf{x}_{k-1}^i$ );
for  $i \leftarrow 1$  to  $N$  do
   $\mathbf{x}_k^i \leftarrow$  sample from  $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$ ;
 $\rho \leftarrow$  root-finding method;
if  $\rho > 0$  then
  for  $i \leftarrow 1$  to  $N$  do
     $w_k^i \leftarrow$  particle weights by (3.4);
  for  $i \leftarrow 1$  to  $N$  do
     $\mathbf{x}_k^i \leftarrow$  systematic resampling
  for  $i \leftarrow 1$  to  $N$  do
     $\mathbf{x}_k^i \leftarrow$  smoothing by (3.6);
if  $\rho < 1$  then
  for  $i \leftarrow 1$  to  $N$  do
     $[\mathbf{z}_k^i, \mathbf{x}_k^i] \leftarrow$  sample from  $p(\mathbf{z}_k|\mathbf{x}_k)^{1-\rho}$ ;
   $\hat{S}^{\mathcal{X}} \leftarrow$  estimator of  $S^{\mathcal{X}}$  from  $[\mathbf{z}_k^i, \mathbf{x}_k^i]$  by (2.34).  $S^{\mathcal{X}}$  is defined by (2.35);
  for  $i \leftarrow 1$  to  $N$  do
     $\mathbf{x}_k^i \leftarrow \hat{S}^{\mathcal{X}}(\mathbf{z}_k^i, \mathbf{x}_k^i)$ ;
     $\mathbf{x}_k^i \leftarrow \hat{S}^{\mathcal{X}}(\mathbf{z}_k^*, \mathbf{x}_k^i)^{-1}$ ;
return  $\mathbf{x}_k^i$ 

```

3.2.1.1 Univariate Non-Stationary Growth (UNGGM) Model

The UNGM model can be expressed as

$$(3.8) \quad \begin{aligned} x_k &= 0.5x_{k-1} + 25 \frac{x_{k-1}}{1 + x_{k-1}^2} + 8 \cos(1.2(k-1)) + v_{k-1} \\ z_k &= x_k + w_k \end{aligned}$$

where $v_k \sim \mathcal{N}(0, 1)$, $w_k \sim \mathcal{N}(0, 2.5)$ are zero-mean Gaussian noise. In our experiments, we set the total number of steps to 100 and repeat the simulations 100 times. The initial state follows $N(20, 1^2)$.

3.2.1.2 Henon Map

Different from the other models, for the Henon map, the number of steps is set to 1. This helps to focus on a single Bayesian assimilation update. Hence, the complication related to varying non-Gaussianity along cycled steps can be eliminated.

From [31], the prior is the density two-dimensional vector \mathbf{x}_1 whose samples are obtained by pushing samples of normal two-dimensional vector \mathbf{x}_0 forward through the Henon map, i.e.

$$(3.9) \quad \begin{aligned} \mathbf{x}_1(1) &= 1 - 1.4\mathbf{x}_0(1)^2 + \mathbf{x}_0(2), \\ \mathbf{x}_1(2) &= 0.3\mathbf{x}_0(1) \end{aligned}$$

where $\mathbf{x}_0(1) \sim \mathcal{N}(0, 1^2)$, $\mathbf{x}_0(2) \sim \mathcal{N}(0, 0.1^2)$. The true values of $\mathbf{x}_1(1)$ and $\mathbf{x}_1(2)$ are set to -4 and 0.6. 1000 Monte Carlo runs were simulated in the experiments.

3.2.1.3 Lorenz-63

The Lorenz-63 Model is a 3-dimensional nonlinear state-space model, and its state vector is represented by $\mathbf{x}_t = (\mathbf{x}_t(1), \mathbf{x}_t(2), \mathbf{x}_t(3))$. The dynamics of the model can be described by

$$(3.10) \quad \begin{aligned} \frac{d\mathbf{x}_t(1)}{dt} &= -10\mathbf{x}_t(1) + 10\mathbf{x}_t(2) \\ \frac{d\mathbf{x}_t(2)}{dt} &= -\mathbf{x}_t(1)\mathbf{x}_t(3) + 28\mathbf{x}_t(1) - \mathbf{x}_t(2). \\ \frac{d\mathbf{x}_t(3)}{dt} &= \mathbf{x}_t(1)\mathbf{x}_t(2) + \frac{8}{3}\mathbf{x}_t(3) \end{aligned}$$

The ODE system above is integrated by a fourth-order explicit Runge-Kutta method and a constant stepsize of $\Delta t = 0.05$. Then at each integration step, we add a Gaussian process noise, $\mathcal{N}(0, 10^{-4}\mathbf{I}_3)$, to the state.

For the observation model, we assume the states are measured every $\Delta t_{obs} = 0.5$ time units, then the measurement model can be represented as

$$(3.11) \quad \mathbf{z}_k = \mathbf{x}_k + \mathbf{v}_k$$

where $\mathbf{x}_k = \mathbf{x}_t|_{t=k\Delta t_{obs}}$ and $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, 4\mathbf{I}_3)$ is zero-mean white Gaussian observation noise.

For the initialisation, the true state is set by sampling from $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_3)$. The true model runs 6000-time steps to generate the states, and the observations are produced by the measurement models. The initial ensemble is produced by a spin-up stage. N samples are obtained from the initial condition $\mathcal{N}(\mathbf{0}, \mathbf{I}_3)$, and the stochastic EnKF is run for the first 2000 steps. Then the Lorenz-63 system reaches a stable stage, and the produced particles are used as initial samples. In the last 4000 steps, several filters are run. Only the filter results for the last 2000 steps are used for the analysis of the filter quality, and the middle 2000 steps are used for eliminating the influence of switching between EnKF and other filters at the end of the 2000th iteration.

3.2.1.4 Target Tracking With Heavy-Tailed Measurement Noise

In this subsection, we describe a nonlinear target tracking model [35, 36]. As it is shown below, the state of the target \mathbf{x}_k is a four-dimensional vector, where p_k^{east} and p_k^{north} are the map coordinates of the object, v_k^{abs} is the value of velocity, and φ_k is the direction of movement.

$$(3.12) \quad \mathbf{x}_k = \begin{bmatrix} p_k^{east} \\ p_k^{north} \\ v_k^{abs} \\ \varphi_k \end{bmatrix}$$

The dynamics of the target tracking model can be described by

$$(3.13) \quad \mathbf{x}_k = \begin{bmatrix} p_{k-1}^{east} + \Delta t_{obs} * \cos(\varphi_{k-1}) * v_{k-1}^{abs} \\ p_{k-1}^{north} + \Delta t_{obs} * \sin(\varphi_{k-1}) * v_{k-1}^{abs} \\ v_{k-1}^{abs} \\ \varphi_{k-1} \end{bmatrix} + \mathbf{w}_{k-1}$$

where Δt_{obs} is the time interval from the current state to the next state, and \mathbf{w}_{k-1} is the heavy-tailed process noise. Different from [35, 36], in our experiment, the heavy-tailed noise is approximated by a GM as is used in various other target tracking models [22, 40], and can be expressed as

$$(3.14) \quad \mathbf{w}_k \sim \begin{cases} N(0, \mathbf{Q}) & \text{with probability } 0.85 \\ N(0, \rho\mathbf{Q}) & \text{with probability } 0.15 \end{cases}$$

where \mathbf{Q} is the covariance matrix, and ρ is a diagonal matrix used to enlarge \mathbf{Q} . Then, the measurement equation is given by

$$(3.15) \quad \mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k$$

where $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R})$ is the measurement noise with covariance \mathbf{R} and

$$(3.16) \quad \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

is the transformation matrix. Following the measurement model, at every time instant, the position coordinates of the target, p_k^{east} and p_k^{north} , are observed. For the complete definition, we also need to specify the observation interval Δt_{obs} , the values for the initial state \mathbf{x}_0 with the covariance matrix \mathbf{C}_0 , the processing noise covariance \mathbf{Q} , and

the measurement noise covariance \mathbf{R} . In the experiment, without loss of generality, the target starts to move from the initial point and the real initial state \mathbf{x}_0 is set to $\begin{bmatrix} 0\text{m} & 0\text{m} & 30\text{m/s} & 0\text{rad} \end{bmatrix}$, where the initial speed of 30m/s is a reasonable value for a vehicle. For simplicity, we assume the initial state subject to a Gaussian distribution with the covariance \mathbf{C}_0 where the deviation about the position, velocity, and direction is set to $\begin{bmatrix} 10\text{m} & 10\text{m} & 3\text{m/s} & \pi/10\text{rad} \end{bmatrix}$, so we have

$$\mathbf{C}_0 = \text{diag} \begin{bmatrix} 100\text{m}^2 & 100\text{m}^2 & 9(\text{m/s})^2 & \pi^2/100\text{rad}^2 \end{bmatrix}.$$

In our experiments, we set

$$\begin{aligned} \Delta t_{obs} &= 1, \\ \mathbf{Q} &= \text{diag} \begin{bmatrix} 0.01\text{m}^2 & 0.01\text{m}^2 & 0.01(\text{m/s})^2 & (\pi/90)^2\text{rad}^2 \end{bmatrix}, \\ \boldsymbol{\rho} &= \text{diag} \begin{bmatrix} 100 & 100 & 100 & 900 \end{bmatrix}, \\ \mathbf{R} &= \text{diag} \begin{bmatrix} 9\text{m}^2 & 9\text{m}^2 \end{bmatrix}. \end{aligned}$$

The whole tracking process lasts for 120 seconds and 50 Monte Carlo runs were simulated.

3.2.2 Benchmark Filters

In our experiments, the proposed hybrid PSMF filter is compared to 6 state-of-the-art filters, including the classic PF, EnKF, ESRF, SMF and two-hybrid filters of SIR-ESRF and GMM-EnKF. In this section, we introduce the configuration of these filters.

For PSMF-L and PSMF-NL, we set both the smoothing parameter β_{inf} to 0.2. Following [87], linear and nonlinear transport maps take the "linear" and "linear+2 RBFs" forms, respectively. The Matlab codes for the transport map are available at: <https://github.com/map-filters/stochasticMaps>. Also, the SMF part in the proposed hybrid filter processes the observations at a given time sequentially.

For a fair comparison, other filters' configurations follow that of PSMF according to their structures. First, systematic resampling is performed in PF, SIR-ESRF and GMM-EnKF. Second, the same smoothing step is adopted in PF. Third, the map design configurations are the same as in SMF. Finally, the observations are sequentially processed in EnKF, ESRF, SMF, and the ESRF/EnKF parts in hybrid filters.

3.2.3 Performance Evaluation in Relation to Ensemble Size

In the first set of experiments, we numerically investigated the performance of filters under different ensemble sizes over a range of state-space models. The number of particles used was successively $N = [20, 40, 60, 100, 200, 400, 600]$. For the hybrid filters, the parameter θ has influences on their filtering performance. In this experiment, the value of θ was taken from a vector $[0.001, 0.002, 0.004, 0.006, 0.008, 0.01, 0.02, 0.04, 0.06, 0.08, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.92, 0.94, 0.96, 0.98, 0.99, 0.992, 0.994, 0.996, 0.998, 0.999]$, and we selected the best performing θ and its corresponding results as the final results. The coordinates in the vector range from 0.001 to 0.999. In the scales near 0 and near 1, a smaller sampling interval is chosen. The reason behind this is that for some of the state-space models, the performance of the hybrid filters is more sensitive to the change of θ within those two scales. The minimum θ in the vector can ensure the effective numbers are less than 1, and the maximum larger than $N - 1$, which can ensure the best parameter values are in the tested range. Alternatively, we can use a root-finding method to search the best performing hybrid filters and their corresponding values of θ [31]. The root-mean-square error (RMSE) was used to analyse the performance of each filter. For the UNGM and Lorenz-63, estimations on all variables were used to calculate RMSE values. For the Henon map model, following [31], the variable $\mathbf{x}_1(1)$ and $\mathbf{x}_1(2)$ were analysed separately whilst the location estimation precision was considered for the target tracking model. In addition, the average continuous ranked probability score (CRPS) is also computed. Different from RMSE, it quantifies the spread of the ensemble. Lower CRPS indicates that the ensemble concentrates around the true values, and is more precise for state estimation.

3.2.3.1 Evaluating PSMF vs. Traditional Filters

RMSE comparisons between the PSMF and the traditional filters are plotted in Figure 3.1. When compared to the PF for a small number of particles ($N < 100$), the proposed approach provides improved results in all cases. For large ensemble sizes, although PSMF still yields better tracking performance under higher dimensional models, similar estimations are obtained under the one-dimension model. These comparisons indicate that different from the classic PF, the proposed hybrid filter can effectively alleviate particle degeneracy due to employing the parameter θ . Note however that in one-dimensional cases, because of the narrow sampling space, for medium and large numbers of particles, the classic PF does not suffer from particle degeneracy seriously, and PSMF does not

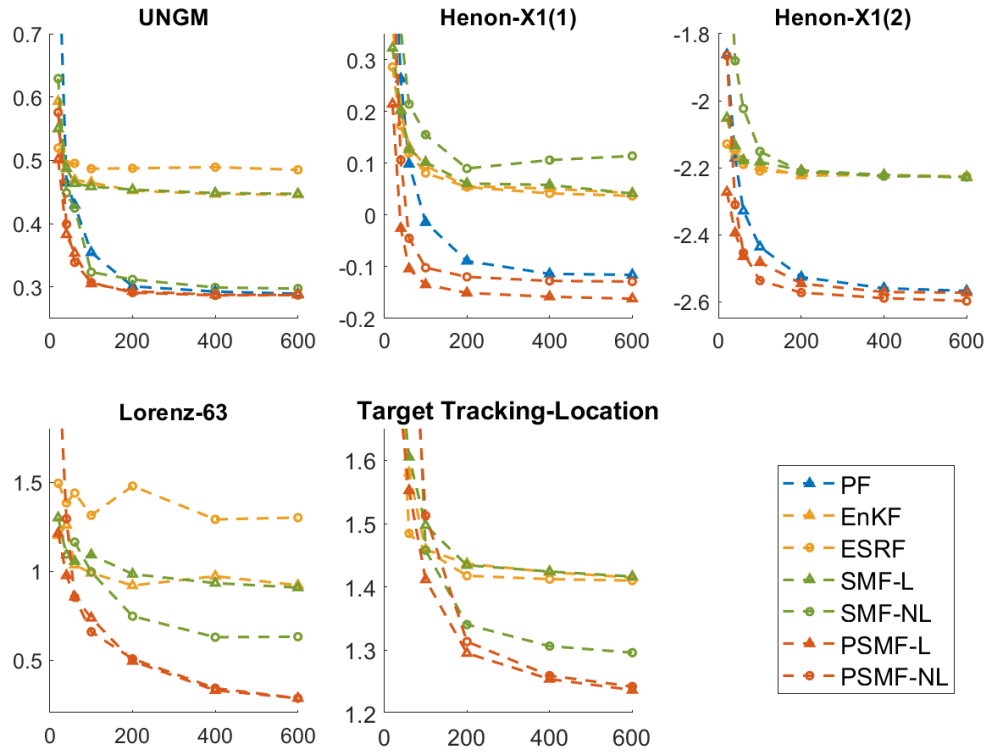


Figure 3.1: The comparison of the proposed PSMF to the traditional filters under different particle numbers. The horizontal axis represents the number of particles, and the vertical axis represents the $\log(\text{RMSE})$.

make a significant difference in these cases.

For small N , the PSMF has similar results with the EnKF/ESRF, whereas when increasing the number of particles, the PSMF starts to show a considerable gain for models. This shows that with few number of samples, the PSMF utilises a larger θ value to maintain the effective number of particles, so that ρ tends to be smaller leading to the PSMF performing closely to the SMF. By contrast, with large N , the PSMF adopts a smaller θ , and ρ increases. Then, the PF part occupies a larger assimilation proportion, and more nonlinear information can be extracted. On the other hand, due to their limited flexibility, the EnKF/ESRF does not yield better results with an increased number of particles.

Specifically speaking for the linear and nonlinear versions of the proposed approach, for small N , the PSMF-L yields better performance compared to the PSMF-NL. In most cases, with an increase in the number of particles, the advantages of the PSMF-L

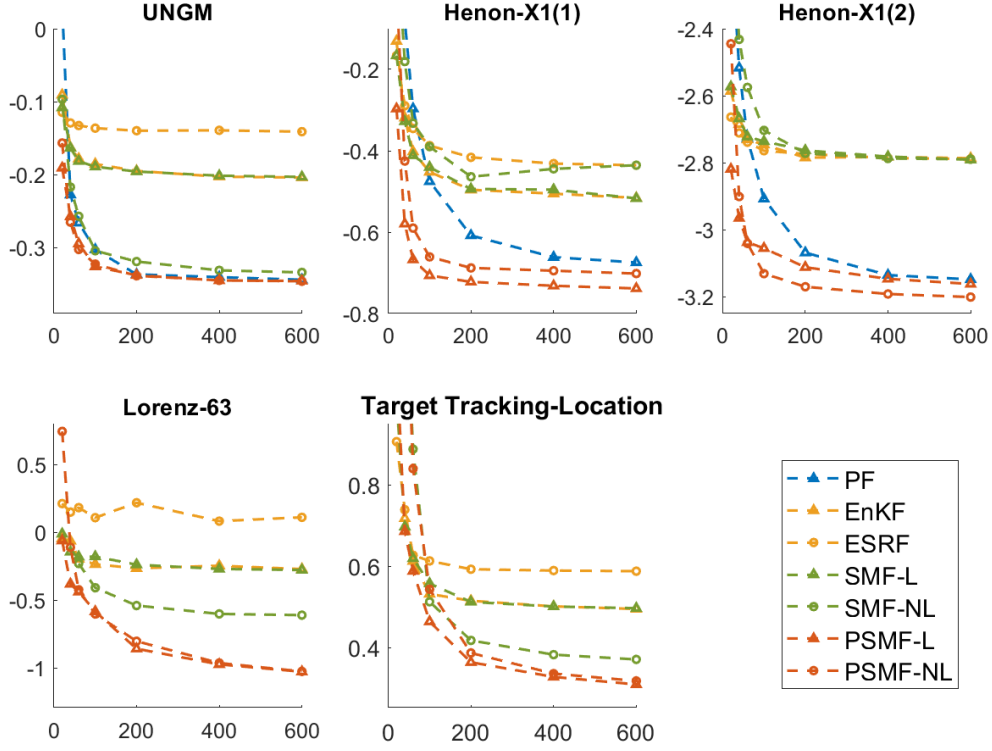


Figure 3.2: The comparison of the proposed PSMF to the traditional filters under the different number of particles. The horizontal axis represents the number of particles, and the vertical axis represents the $\log(\text{CRPS})$.

gradually reduce. The reasons behind this are related to the quality of transport maps. From [87], we can recall that a linear map has a more robust performance with small N but can have limited flexibility. By contrast, a nonlinear map holds higher non-linearity but might yield unacceptable variances without a sufficient number of particles. As a result, for small N , nonlinear maps cause a large variance, and the PSMF-NL performs worse than its linear counterpart, the PSMF-L. For large N , the variance produced by the nonlinear maps can be suppressed and the difference between PSMF-L and PSMF-NL becomes smaller.

The results above do not imply that the PSMF-L can always achieve better results than PSMF-NL. From the results under the variable $\mathbf{x}_1(2)$ of the Henon map, PSMF-NL provides better results. As mentioned before, compared with the nonlinear maps, linear maps have very limited flexibility, which causes poor estimation results. In some cases, a large θ is needed to maintain the effective number of particles, and then the value

of ρ decreases. Consequently, the SMF occupies a larger assimilation part and needs to perform more complex transportations. If the bias caused by the linear transport approximation is larger than that of nonlinear-map variance, the PSMF-NL can perform better than the PSMF-L.

To complete this section, CRPS comparisons between the PSMF and the traditional filters are also plotted in Figure 3.2 and they follow the same characteristics similar to the RMSE results in Figure 3.1. This suggests that our new hybrid filter shows advantages over the traditional filters in both mean estimations and ensemble concentration.

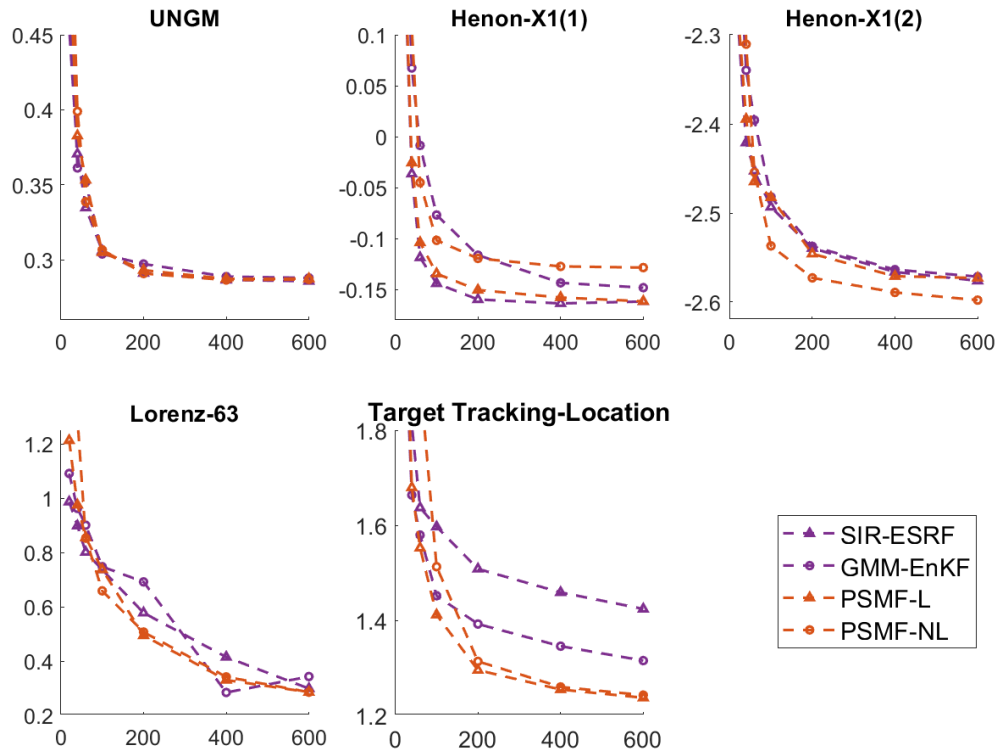


Figure 3.3: The comparison of the PSMF to the other hybrid filters under the different number of particles. The horizontal axis represents the number of particles, and the vertical axis represents $\log(\text{RMSE})$.

3.2.3.2 PSMF vs. Hybrid Filters

RMSE comparisons between the PSMF and other hybrid filters are plotted in Figure 3.3.

When compared to the SIR-ESRF, the proposed PSMF performs worse when the particle number is small. In most cases, for an increasing number of particles, the

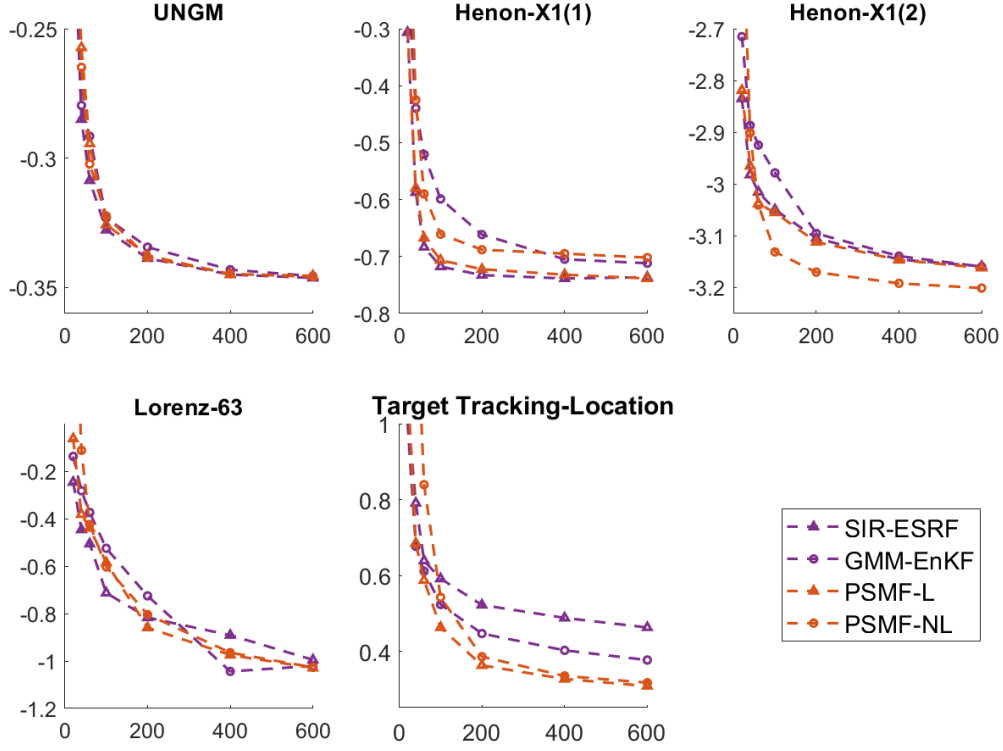


Figure 3.4: The comparison of the proposed PSMF to the traditional filters under the different number of particles. The horizontal axis represents the number of particles, and the vertical axis represents $\log(CRPS)$.

proposed PSMF's performance achieved is closer to or better than that of SIR-ESRF.

The difference between the performance of the SIR-ESRF and that of PSMF is primarily due to their distinct strategies for alleviating particle degeneracy. In the PSMF, systematic resampling with a smoothing step is adopted. By contrast, SIR-ESRF relies on the mean-preserving random orthogonal transformation which resamples the ensemble during the ESRF assimilation stage. This resampling technique performs well under medium-level nonlinear models. However, since it is based on the Gaussian approximation, when the posterior distribution gets further from Gaussian, more errors will be introduced. In all the experiments, for small N , the bias caused by the systematic resampling and transport map variances is larger than that of the Gaussian approximation in SIR-ESRF, which thus determines the proposed PSMF to achieve closer or worse results. However, for medium and large N , the bias from the systematic resampling and transport map variance is less significant. As a result, it becomes smaller than the error

from the resampling in the SIR-ESRF.

For small N , the PSMF and GMM-EnKF achieve considerably lower performance compared to the SIR-ESRF. However, the GMM-EnKF achieves worse results than the proposed PSMF for a medium number of particles. When increasing the number of particles, its performance gradually approaches that of PSMF, although under the nonlinear non-Gaussian target tracking model, for $N = 600$, the PSMF still offers obvious advantages. These results are caused by the structure of the GMM-EnKF. Different from the other hybrid filters, the GMM-EnKF implements the EnKF assimilation first, which is based on the Gaussian assumption for the prior distributions. In all the state-space models in this chapter, all the measurement models are linear and Gaussian, which means that the posterior has a closer form to a Gaussian than the prior. Thus, implementing ensemble-based filters first introduces more errors. For small N , the EnKF update occupies a large part to keep the ESS . As a result, the Gaussian approximation introduces more errors. For large N , the GMM-EnKF just needs a small θ value to ensure the ESS is large enough. Consequently, a smaller ρ will be selected and the GMM occupies a large proportion. Hence, more nonlinear transform information between the prior and posterior can be obtained. However, the 4-dimensional target tracking model has a larger sampling space and needs more effective particles. So, even for $N = 600$, the EnKF occupies a larger assimilation proportion than the other cases. As a result, the GMM-EnKF is significantly outperformed by the proposed PSMF.

As in the previous section, we also provide comparisons between the PSMF and the other hybrid filters in terms of CRPS and these are plotted in Figure 3.4. The results are consistent with the RMSE results in Figure 3.3. This means that the differences among the hybrid filters in the mean estimation can be extended to the ensemble concentration. Finally, for further objective evaluation, numerical results obtained for both RMSE and CRPS are also shown in Table 3.1 and Table 3.2, respectively. Due to the limited space, a sparse selection of ensemble sizes, $[20, 60, 200, 600]$, is provided. Results indicate that in most cases, hybrid filters work better than traditional filters. Also, SIR-ESRF tends to perform better than the others with few particles, while PSMF obtains robust performance with large ensemble sizes.

3.2.4 The effect of θ on filter performance

In the second set of experiments, we investigated the influence of θ on the hybrid filters. We used the same values of θ as in subsection 3.2.3, i.e. between 0.001 and 0.999. The numbers of particles used were 20, 200, and 600.

R	N	F1	F2	F3	F4	F5	F6	F7	F8	F9
M1	20	2.375	1.81	1.681	1.733	1.876	1.576	1.647	1.651	1.778
	60	1.536	1.598	1.641	1.59	1.529	1.398	1.422	1.424	1.403
	200	1.351	1.572	1.628	1.574	1.366	1.338	1.346	1.341	1.338
	600	1.335	1.562	1.624	1.564	1.346	1.331	1.334	1.333	1.332
M2	20	1.821	1.446	1.331	1.38	2.008	1.227	1.323	1.239	1.662
	60	1.103	1.14	1.127	1.135	1.239	0.888	0.991	0.901	0.956
	200	0.915	1.057	1.055	1.063	1.094	0.852	0.89	0.86	0.887
	600	0.891	1.043	1.038	1.042	1.121	0.85	0.862	0.851	0.879
M3	20	0.155	0.128	0.119	0.129	0.25	0.102	0.114	0.103	0.155
	60	0.097	0.113	0.112	0.113	0.132	0.086	0.091	0.085	0.086
	200	0.08	0.108	0.109	0.11	0.11	0.079	0.079	0.078	0.076
	600	0.077	0.108	0.108	0.108	0.108	0.076	0.076	0.076	0.074
M4	20	20.361	3.323	4.451	3.667	NaN	2.681	2.977	3.361	9.33
	60	20.229	2.815	4.214	2.872	3.198	2.228	2.46	2.35	2.346
	200	17.282	2.506	4.383	2.673	2.11	1.782	1.997	1.639	1.66
	600	13.533	2.511	3.671	2.477	1.88	1.348	1.409	1.33	1.33
M5	20	996.963	7.788	6.745	8.551	59.663	8.549	8.136	9.214	912.3
	60	690.83	4.856	4.413	4.979	7.857	5.14	4.853	4.723	6.756
	200	458.637	4.207	4.127	4.196	3.821	4.519	4.024	3.652	3.719
	600	262.829	4.116	4.096	4.122	3.655	4.153	3.726	3.445	3.465

Table 3.1: RMSE results of the filters under different models. R and N represent RMSE and the ensemble size, respectively. M1:M5 represent UNGM, Henon-u, Henon-v, Lorenz-63, and Target tracking, respectively. F1:F9 represent PF, EnKF, ESRF, SMF-L, SMF-NL, SIR-ESRF, GMM-EnKF, PSMF-L, PSMF-NL. The first and second minimal RMSEs under different particle numbers and models are highlighted in red and blue bold fonts, respectively.

In Figure 3.5, the influence of parameter θ is evaluated, for an ensemble size of $N = 20$. It can be observed that, in general, the best results are achieved by the SIR-ESRF. As was mentioned in the previous sections, the reason behind this is that the resampling technique of the SIR-ESRF, i.e. the mean-preserving random orthogonal transformation, has a more robust performance for small N . In addition, the large variance of nonlinear maps for a small number of particles makes the proposed nonlinear approach inefficient in this case.

The PSMF-L and GMM-EnKF achieve inconsistent results and lower performance than the SIR-ESRF in most cases. This is due to the known drawbacks of these two filters: when using few particles, the systematic resampling technique in PSMF-L is not as robust as that in SIR-ESRF. By contrast, the GMM-EnKF runs by first performing

C	N	F1	F2	F3	F4	F5	F6	F7	F8	F9
M1	20	1.031	0.913	0.892	0.898	0.908	0.82	0.843	0.826	0.855
	60	0.767	0.837	0.876	0.834	0.774	0.734	0.747	0.745	0.739
	200	0.714	0.823	0.87	0.822	0.727	0.713	0.716	0.714	0.713
	600	0.709	0.816	0.868	0.816	0.716	0.707	0.708	0.708	0.708
M2	20	1.427	0.878	0.847	0.847	1.129	0.736	0.812	0.743	1.002
	60	0.743	0.669	0.708	0.663	0.717	0.505	0.594	0.513	0.554
	200	0.544	0.609	0.66	0.61	0.629	0.481	0.516	0.486	0.503
	600	0.509	0.597	0.647	0.596	0.647	0.479	0.491	0.478	0.496
M3	20	0.118	0.075	0.07	0.076	0.136	0.059	0.066	0.06	0.087
	60	0.065	0.065	0.065	0.066	0.076	0.049	0.054	0.048	0.048
	200	0.046	0.062	0.062	0.063	0.063	0.045	0.045	0.045	0.042
	600	0.043	0.062	0.062	0.061	0.061	0.042	0.042	0.042	0.041
M4	20	9.28	0.954	1.235	0.987	NaN	0.781	0.871	0.939	2.102
	60	9.355	0.82	1.198	0.834	0.794	0.603	0.688	0.644	0.654
	200	6.99	0.766	1.242	0.787	0.582	0.442	0.484	0.423	0.447
	600	4.466	0.762	1.116	0.757	0.542	0.37	0.36	0.357	0.358
M5	20	378.726	2.738	2.474	2.773	14.909	2.83	2.73	3.043	338.948
	60	224.118	1.844	1.871	1.857	2.429	1.895	1.843	1.8	2.315
	200	97.731	1.673	1.808	1.668	1.518	1.686	1.564	1.438	1.472
	600	36.274	1.64	1.799	1.643	1.449	1.589	1.459	1.362	1.374

Table 3.2: CRPS results of the filters under different models. C and N represent CRPS and the ensemble size, respectively. M1:M5 represent UNGM, Henon-u, Henon-v, Lorenz-63, and Target tracking, respectively. F1:F9 represent PF, EnKF, ESRF, SMF-L, SMF-NL, SIR-ESRF, GMM-EnKF, PSMF-L, PSMF-NL. The first and second minimal RMSEs under different particle numbers and models are highlighted in red and blue bold fonts, respectively.

the EnKF component, which leads to large errors.

In Figure 3.6, the influence of the parameter θ is evaluated for $N = 200$. It can be seen that for the first 3 models, for the same reason as above, the GMM-EnKF yields the worst results. It is clear that the proposed PSMF-NL achieves the best performance for large θ values (close to 1). The improved results benefit from the larger flexibility of the nonlinear maps and reduced map variances in the case of a medium number of particles. However, when θ is small, the advantage of PSMF-NL is not obvious anymore, the reason being that in such case, the values of ρ tend to be large, and the SMF part occupies a small proportion of the assimilation process. Consequently, there is no need for a flexible nonlinear map to complete the assimilation. Also, we should notice that the performance of the PSMF-NL is just slightly worse than that of PSMF-L in this case. The

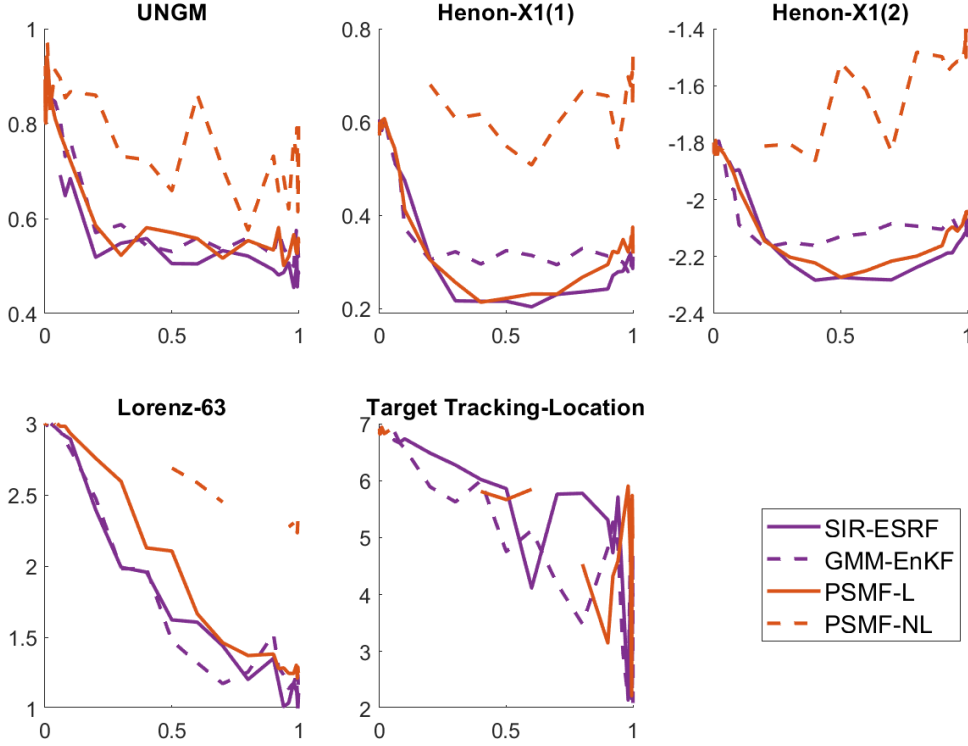


Figure 3.5: The comparison among hybrid filters with different values of parameter θ under different models. The ensemble size N is set to 20. Horizontal axis represents the value of θ , and vertical axis represents the estimation $\log(\text{RMSE})$.

small effective number of particles caused by small θ do not cause large nonlinear map variances. Because of our smoothing strategy, the diversity of particles can be ensured and the variance can be suppressed.

For the first 3 models, the PSMF-L and SIR-ESRF achieve comparable results with lower performance compared to the proposed PSMF-NL. The disadvantages of the SIR-ESRF stem from the Gaussian assumption of the resampling techniques, whilst the disadvantages of the PSMF-L are due to the lower flexibility of linear maps.

For the last model (4D tracking), the results are slightly different. First, the SIR-ESRF performs relatively worse than the proposed PSMF-L since the posterior distributions under this model are far from the standard Gaussian assumption. Second, when θ is close to 1, the advantage of the proposed PSMF-NL can still be seen. Nevertheless, on a large scale, the PSMF-NL performs worse than its linear counterpart. This is due to the increased variance of the nonlinear map in this case. According to the structure of

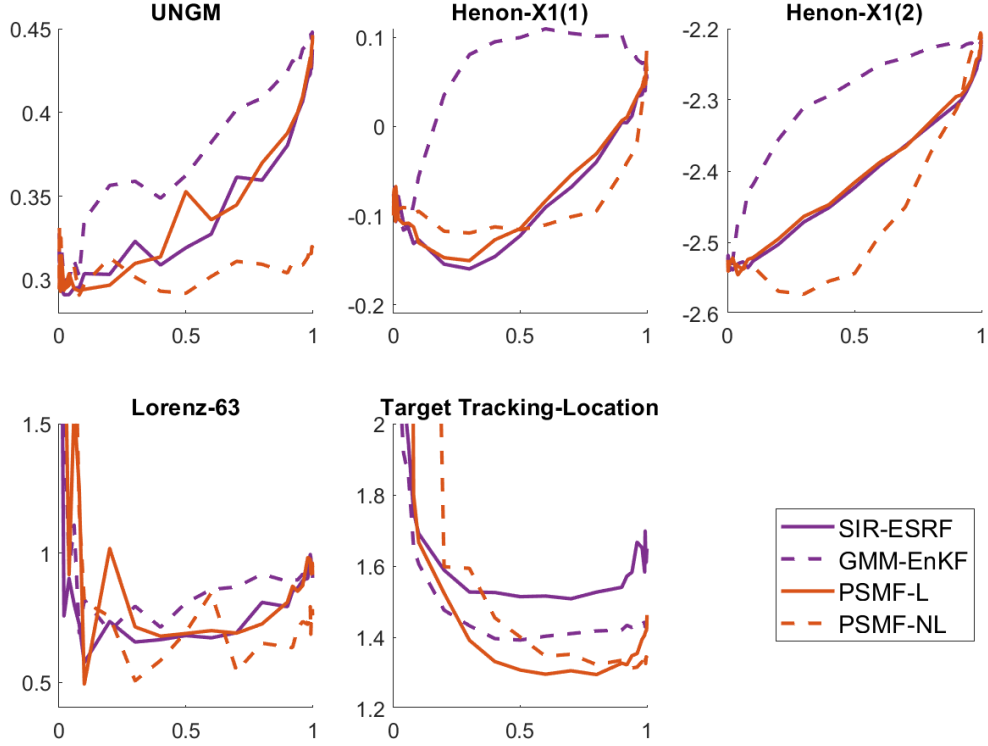


Figure 3.6: The comparison among hybrid filters with different values of parameter θ under different models. The ensemble size N is set to 200. Horizontal axis represents the value of θ , and vertical axis represents the estimation $\log(\text{RMSE})$.

the triangular maps, it is known that the number of map parameters increases with the dimension of models. As a result, a medium number of particles cannot effectively suppress the variance of nonlinear maps that causes PSMF-NL to perform worse than the linear PSMF-L.

Finally, in Figure 3.7, the influence of parameter θ is evaluated, for $N = 600$. The results are similar to those obtained for $N = 200$, with some slight differences. It can be seen that For large N , the proposed PSMF-NL benefits more from the increase in the number of particles. For a larger scale of values of θ , it outperforms the others for the first three models. Even under the 4D target tracking model, the variance of nonlinear maps can be suppressed more efficiently, and its performance becomes closer to that of the PSMF-L.

We can also notice that under the target tracking model, when $0.2 < \theta < 0.8$, the location estimation $\log(\text{RMSE})$ of the proposed PSMF varies very little. We attribute

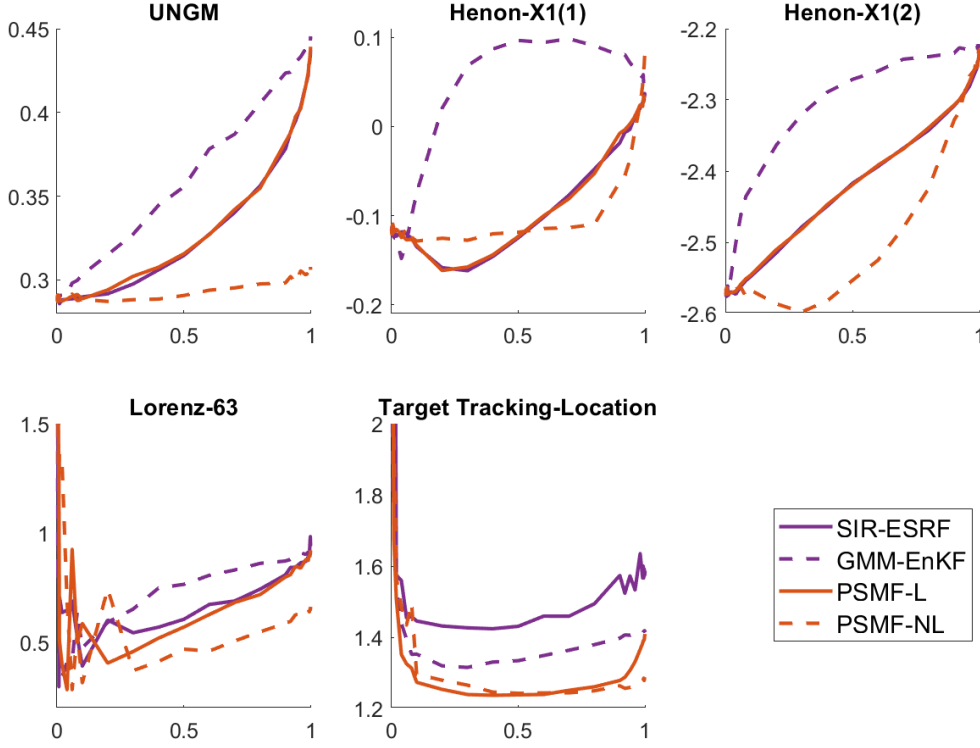


Figure 3.7: The comparison among hybrid filters with different values of parameter θ under different models. The ensemble size N is set to 600. Horizontal axis represents the value of θ , and vertical axis represents the estimation $\log(\text{RMSE})$.

this to the fact that the transformation between a prior distribution and corresponding posterior distribution is not particularly complex under this model. When θ changes between 0.2 and 0.8, the assimilation proportion of PF changes as well. However, the SMF can always complete the remaining assimilation update and the final estimation results remain stable. When θ is very close to 1, transport maps cannot extract all the remaining nonlinear information and the performance is reduced. By contrast, when θ is very close to zero, the PSMF becomes closer to the classical PF and hence suffers from particle degeneracy.

3.2.5 The effect of the smoothing step on filter performance

In this section, the influence of the smoothing step on the filtering performance of PSMFs is investigated. The parameter set is the same as in the first experiment, and the experiment results are shown in Figure 3.8. Under most models, the smoothing step

does not influence the filtering performance. However, under Lorenz-63, with a large number of particles, PSMF performs better than PSMF-nsm. The reason is that with large ensemble sizes, the optimal value of θ is relatively small under Lorenz-63 as shown in Figure 3.7. Thus, ESS is small, which causes large map variances. A smoothing step can increase the diversity of samples, and then the map variance is decreased. Hence, in this case, PSMF works better than PSMF-nsm.

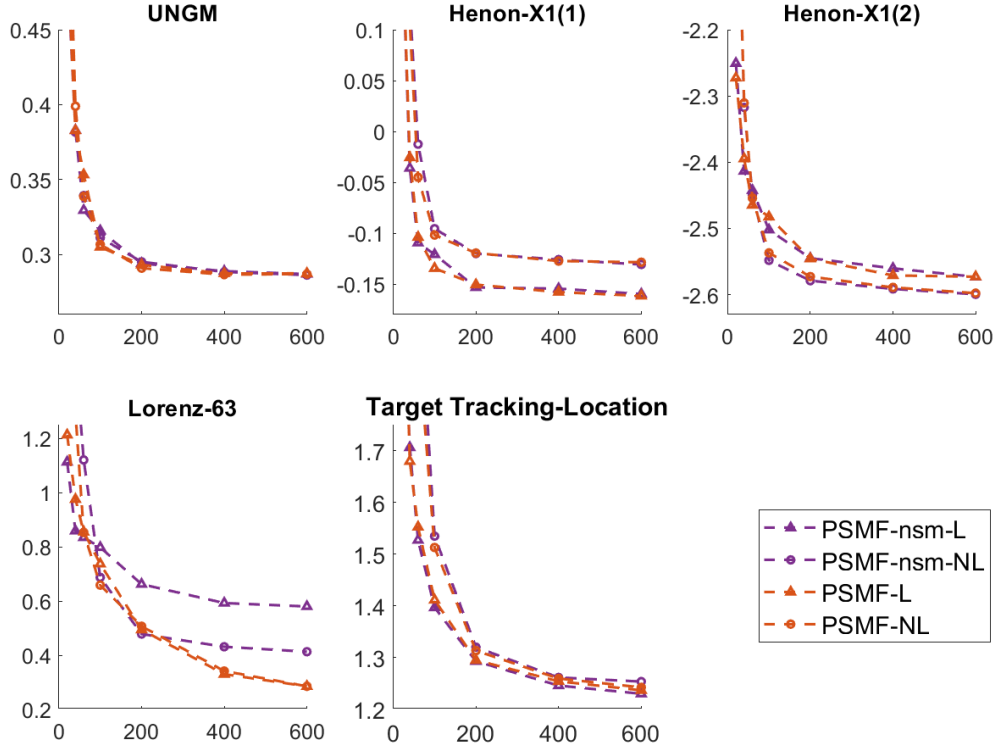


Figure 3.8: The comparison of the proposed PSMFs between with and without the smoothing step under the different number of particles. PSMF-nsm-L and PSMF-nsm-NL represent linear and nonlinear PSMFs without the smoothing step, respectively. The horizontal axis represents the number of particles, and the vertical axis represents $\log(RMSE)$.

3.2.6 Computational Complexity Analysis

For an exhaustive evaluation of our proposed PSMF method, the filter efficiency with the change of particle numbers is also evaluated. To this end, we fix the parameter θ to 0.5, and employ ensemble sizes of [100, 200, 400, 600, 1000]. Small particle numbers

are not considered because some filters diverge with few samples and then the precise computational cannot be obtained. The rest of the parameters are the same as in the first set of experiments presented above. The computational time is shown in Figure 3.9.

Compared with traditional methods, PSMF is less efficient than PF, EnKF and ESRF. However, the comparison between PSMF and SMF does not lead to consistent conclusions, being dependent on the state-space models investigated. Specifically, under the one-dimensional UNGM model, the PF part of PSMF can keep the *ESS* high, and the SMF part is omitted in many iterations. Consequently, PSMF is more efficient than SMF. By contrast, under higher-dimensional models, the SMF part of PSMF can be omitted in fewer iterations. Thus, the inclusion of both PF and SMF steps in PSMF slows the filtering process.

Also, GMM-EnKF is more time-consuming than PSMF for all the models. This is due to its *ESS* calculation for searching ρ . Specifically, because GMM-EnKF implements EnKF first, the whole hybrid filter should be implemented to calculate *ESS*. However, for PSMF, only the PF part is implemented to obtain *ESS*, as the SMF step does not change *ESS*. Therefore, GMM-EnKF spends more time on the search of ρ by the root-finding method.

Finally, PSMF has a similar computational load to SIR-ESRF when the ensemble size is small. However, the computational time of SIR-ESRF increases faster than PSMF along the x-axis and becomes larger than that of PSMF when particle numbers are large. This is caused by the mean-preserving random orthogonal transformation in SIR-ESRF, which has nonlinear computational complexity with respect to the particle number as explained in [31]. By contrast, PSMF has linear computational complexity.

3.3 Conclusion

In this chapter, we proposed a novel hybrid filtering approach, the PSMF, which enhances the standard particle filter by using ideas from stochastic map filters. To break the particle degeneracy issue, which is peculiar to PF, systematic resampling followed by a smoothing step was adopted. To analyse the impact of the nonlinearity of transport maps, we presented PSMF-L and PSMF-NL by adopting linear and nonlinear transport maps, respectively.

Two sets of experiments involving four widely employed state-space models were implemented to validate the proposed hybrid filters. In the first experimental setup, we investigated the performance of the PSMF with optimal parameters of θ for different

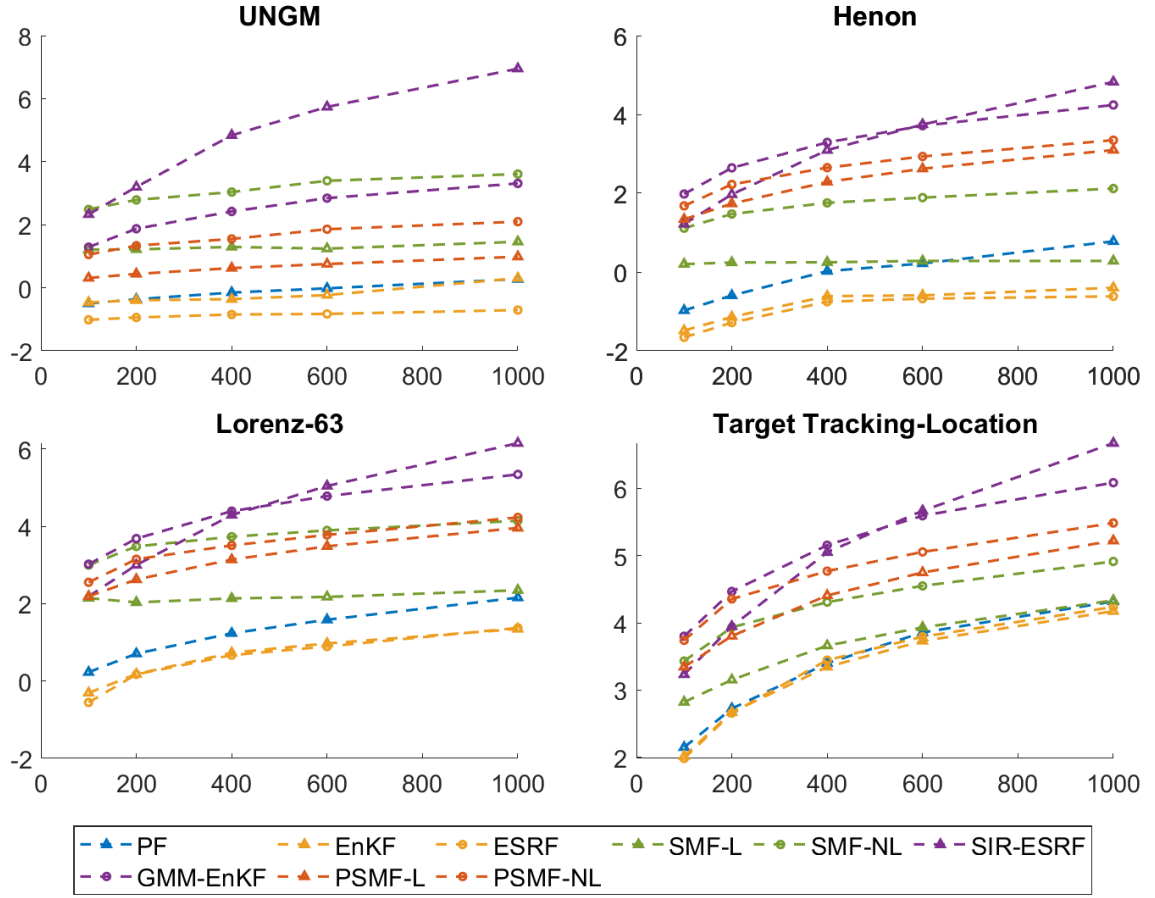


Figure 3.9: The comparison of the proposed PSMF to the other filters under the different number of particles. The horizontal axis represents the number of particles, and the vertical axis represents $\log(\text{time})$.

ensemble sizes. Experiment results showed that for medium and large numbers of particles the linear version of the proposed hybrid filter, PSMF-L, yields better results than the benchmark approaches. Nevertheless, in some cases, the error inherently introduced by the nonlinear map variance was smaller than that due to the linear approximation of the linear map. Consequently, the nonlinear hybrid filter provided better results than the PSMF-L. We should also note that, with a small number of particles, the advantages that the PSMF offers were not obvious when compared to the other filters, since the PSMF suffers errors caused by the systematic resampling and transport map variance.

For the second series of simulations, the relationship between the performance of the PSMF and the parameter θ was investigated. It was demonstrated that in most cases the

PSMF-NL is more tolerant to changes of the parameter θ for medium and large numbers of particles. This owes to the higher flexibility of nonlinear maps. In addition, for small numbers of particles, due to the large nonlinear map variance, the PSMF-NL achieves lower performance with respect to benchmark approaches.

In this work, the PSMF is only tested under low-dimensional model. In contrast, our future work will consider extending the proposed approaches to higher-dimensional problems. Moreover, we will also investigate the deterministic transport map filter (DMF) [87] within the context of the proposed hybrid filters.

ROBUST KALMAN FILTERS BASED ON THE SUB-GAUSSIAN α -STABLE DISTRIBUTION

Although showing robust filtering performance under various nonlinear non-Gaussian models in Chapter 3, the proposed PSMF is computationally expensive due to its reliance on numerous samples. In contrast, this chapter focuses on a new efficient RKF framework under the linear system with heavy-tailed noise. As explained in the chapter 1, due to the requirements for the density, only several heavy-tailed distributions were applied to the RKF framework, and employing more heavy-tailed PDFs to improve the RKFs remains significant. In this chapter, a new robust Kalman filter based on the $SG\alpha S$ distribution [70, 84] is proposed. Also, for the RKF based on the slash distribution (RKF-SL) [41], we present a novel MMSE estimate of the scale function. The rest of the chapter is organised as follows: We begin with the details of the α -stable distribution and its sub-Gaussian case. Also, Section 4.2 provides the formulation of the proposed RKF- $SG\alpha S$, and the improved RKF based on the slash distribution (RKF-SL) is explained in Section 4.3. Further, the proposed framework is tested in target tracking scenarios in Section 4.4, whilst Section 4.5 concludes this work with a summary.

4.1 α -stable Distributions and the Sub-Gaussian Case

In this section, we first introduce the α -stable family of distributions whereby its sub-Gaussian subclass, the SG α S distribution, is also illustrated. Furthermore, the tail behaviour of this subclass is analysed.

In probability theory, the α -stable distribution is a family of probability distributions, which is generally used to model heavy-tailed behaviour and has been employed in various filtering tasks [27, 71, 90]. Particularly, a univariate stable random variable can be defined based on the following stability property [55]:

Definition 4.1. A random variable x , has an α -stable distribution if for any positive numbers A and B , there is a positive number C and a real number D such that

$$Ax_1 + Bx_2 \stackrel{d}{=} Cx + D$$

where x_1 and x_2 are independent copies of x and $\stackrel{d}{=}$ represents equality in distribution.

Also, a stable random variable can be defined based on the generalized Central Limit Theorem (GCLT) [55]:

Definition 4.2. A random variable x has a stable distribution if it has a domain of attraction, i.e., if there is a sequence of i.i.d. random variables y_1, y_2, \dots and sequence of positive numbers $(d_{n'})_{(n' \in \mathbb{N})}$ and real numbers $(D_{n'})_{(n' \in \mathbb{N})}$, such that

$$\frac{y_1 + y_2 + \dots + y_{n'}}{d_{n'}} + D_{n'} \xrightarrow{d} x,$$

where \xrightarrow{d} represents convergence in distribution. The next definition represents a stable variable in terms of its characteristic function [84].

Definition 4.3. A random variable is stable if its characteristic function can be written as

$$\mathbb{E}(\exp(i\theta x)) = \begin{cases} \exp(-\gamma^\alpha |\theta|^\alpha [1 - i\beta (\tan \frac{\pi\alpha}{2})(\text{sign } \theta)] + i\delta\theta), & \alpha \neq 1 \\ \exp(-\gamma |\theta| [1 + i\beta \frac{2}{\pi}(\text{sign } \theta) \ln|\theta|] + i\delta\theta) & , \alpha = 1 \end{cases}$$

where the stability parameter $\alpha \in (0, 2]$, skewness parameter $\beta \in [-1, 1]$, scale parameter $\gamma \in (0, \infty)$ and location parameter $\delta \in \mathbb{R}$.

Furthermore, the multivariate α -stable distribution extends the concept of the univariate stable distribution to high-dimensional cases. Thus, stable random vectors also exhibit the stability property and allow for the GCLT. To define stable random vector, we can simply extend Definitions 4.1 and 4.2, i.e.,

Definition 4.4. A random vector $\mathbf{x} \in \mathbb{R}^d$ is subject to a multivariate stable distribution if and only if there exists a unique finite measure Γ on the unit sphere \mathcal{S}^d , the so-called spectral measure, and a unique vector $\boldsymbol{\delta} \in \mathbb{R}^d$ such that:

1. If $\alpha \neq 1$,

$$\mathbb{E} \left(\exp(i\tilde{\boldsymbol{\Theta}}^T \mathbf{x}) \right) = \exp \left\{ - \int_{\mathcal{S}^d} |\tilde{\boldsymbol{\Theta}}^T \mathbf{s}|^\alpha \left(1 - i \operatorname{sign}(\tilde{\boldsymbol{\Theta}}^T \mathbf{s}) \tan \frac{\pi\alpha}{2} \right) \Gamma(d\mathbf{s}) + i\tilde{\boldsymbol{\Theta}}^T \boldsymbol{\delta} \right\}$$

2. If $\alpha = 1$,

$$\mathbb{E} \left(\exp(i\tilde{\boldsymbol{\Theta}}^T \mathbf{x}) \right) = \exp \left\{ - \int_{\mathcal{S}^d} |\tilde{\boldsymbol{\Theta}}^T \mathbf{s}| \left(1 + i \frac{2}{\pi} \operatorname{sign}(\tilde{\boldsymbol{\Theta}}^T \mathbf{s}) \ln |\tilde{\boldsymbol{\Theta}}^T \mathbf{s}| \right) \Gamma(d\mathbf{s}) + i\tilde{\boldsymbol{\Theta}}^T \boldsymbol{\delta} \right\}$$

However, we cannot directly apply the multivariate stable distribution to RKF frameworks due to the challenge of estimating the spectral measure of its characteristic function [55]. Instead, we focus on a more suitable subclass of the multivariate stable distribution, known as the SG α S distribution. It has a simpler spectral measure, and its characteristic function can be expressed as [91]

$$\mathbb{E} \left[\exp(i\tilde{\boldsymbol{\Theta}}^T \mathbf{x}) \right] = \exp \left[i\tilde{\boldsymbol{\Theta}}^T \boldsymbol{\mu} - \left(\tilde{\boldsymbol{\Theta}}^T \boldsymbol{\Sigma} \tilde{\boldsymbol{\Theta}} \right)^{\frac{\alpha}{2}} \right].$$

As previously noted, the SG α S distribution is also a special case of the Gaussian scale mixture (GSM) distribution which can be written as

$$(4.1) \quad p(\mathbf{x}) = \int_0^{+\infty} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu} + y\boldsymbol{\beta}, \boldsymbol{\Sigma}/\kappa(y)) \pi(y) dy,$$

where $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{\beta}$ is the skewness vector. Also, $\boldsymbol{\Sigma}$ is the scale matrix, $y > 0$ is the mixing parameter, and $\kappa(y)$ and $\pi(y)$ are the scale function and mixing density, respectively. For the SG α S distribution [55], $\boldsymbol{\beta} = \mathbf{0}$, $\kappa(y) = 1/y$ and $\pi(y)$ can be represented as a totally skewed univariate stable distribution, i.e.,

$$(4.2) \quad p(\mathbf{x}) = \int_0^{+\infty} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, y\boldsymbol{\Sigma}) S(y; \alpha/2, 1, \cos(\pi\alpha/2)^{2/\alpha}, 0) dy.$$

Table 4.1 details the parameters of SG α S along with several GSM distributions investigated in this work, where v refers to the dof parameter. Also, a parameter description of the mixing densities is provided in Table 4.2. For simplicity, we only consider the zero-mean symmetric heavy-tailed noise as in previous works [4, 104] where the mean and skewness vectors are zero for both the SG α S and the GSM distributions.

Subsequent to the succinct introduction of the SG α S distribution, we explain its tails behaviour, which is determined by its mixing density. Figure 4.1-(a,b) shows for $\alpha = 2$, the

Table 4.1: Exemplary GSM distributions and their parameters

GSM dis- tribution	Scale function	Mixing density	Constraints
SG α S	$\kappa(y) = y^{-1}$	$\pi(y) = S(y; \alpha/2, 1, \cos(\pi\alpha/2)^{2/\alpha}, 0)$	$y > 0, 0 < \alpha \leq 2$
Student's t	$\kappa(y) = y$	$\pi(y) = G(y; \frac{v}{2}, \frac{v}{2})$	$y > 0, v > 0$
Slash	$\kappa(y) = y$	$\pi(y) = \text{Be}(y; \frac{v}{2}, 1)$	$0 < y < 1, v > 0$
Variance Gamma	$\kappa(y) = y$	$\pi(y) = \text{IG}(y; \frac{v}{2}, \frac{v}{2})$	$y > 0, v > 0$

Table 4.2: Parameter description of the mixing densities

Mixing desity	Definitions
$S(.; \alpha, \beta, \gamma, \delta)$	Univariate stable distribution PDF with the shape parameter α , the skewness parameter β , the scale parameter γ and the location parameter δ .
$G(.; a, b)$	Gamma PDF with the shape parameter a and rate parameter b .
$\text{Be}(.; a, b)$	Beta PDF with the shape parameter a and scale parameter b .
$\text{IG}(.; a, b)$	Inverse-Gamma PDF with the shape parameter a and b .

SG α S distribution follows the Gaussian distribution and its corresponding mixing density is a Dirac delta distribution at 1. Also, for $\alpha < 2$, the SG α S distribution becomes more heavy-tailed and the mixing density is a totally skewed heavy-tailed stable distribution. Besides, smaller α values result in heavier tails as in Figure 4.1-(c,d). Besides, the tail behavior of the stable distribution with $\alpha < 2$ is characterized by a power-law decay rate $c|x|^{-1-\alpha}$, where c is a constant factor, and hence its variances diverge [12]. By comparison, the Gaussian distribution with exponential decay rate $c\exp(-0.5x^2)$ features light tails and finite variances [12]. While the heavy-tailed student's t distribution exhibits a power-law slowly decaying behaviour, its lack of the stability property hinders the theoretical justification for its practical applicability [43].

4.2 RKF-SG α S

Based on the stable distribution explained in the last section, many time-series inference tasks have been explored. For example, based on expectation–maximization (EM) and Markov chain Monte Carlo (MCMC) methods, Bayesian inference methods for random processes with stable noise are presented in [29]. Also, [30] discusses filtering tasks under time-varying autoregressive (TVAR) signal models with α -stable noise, and the

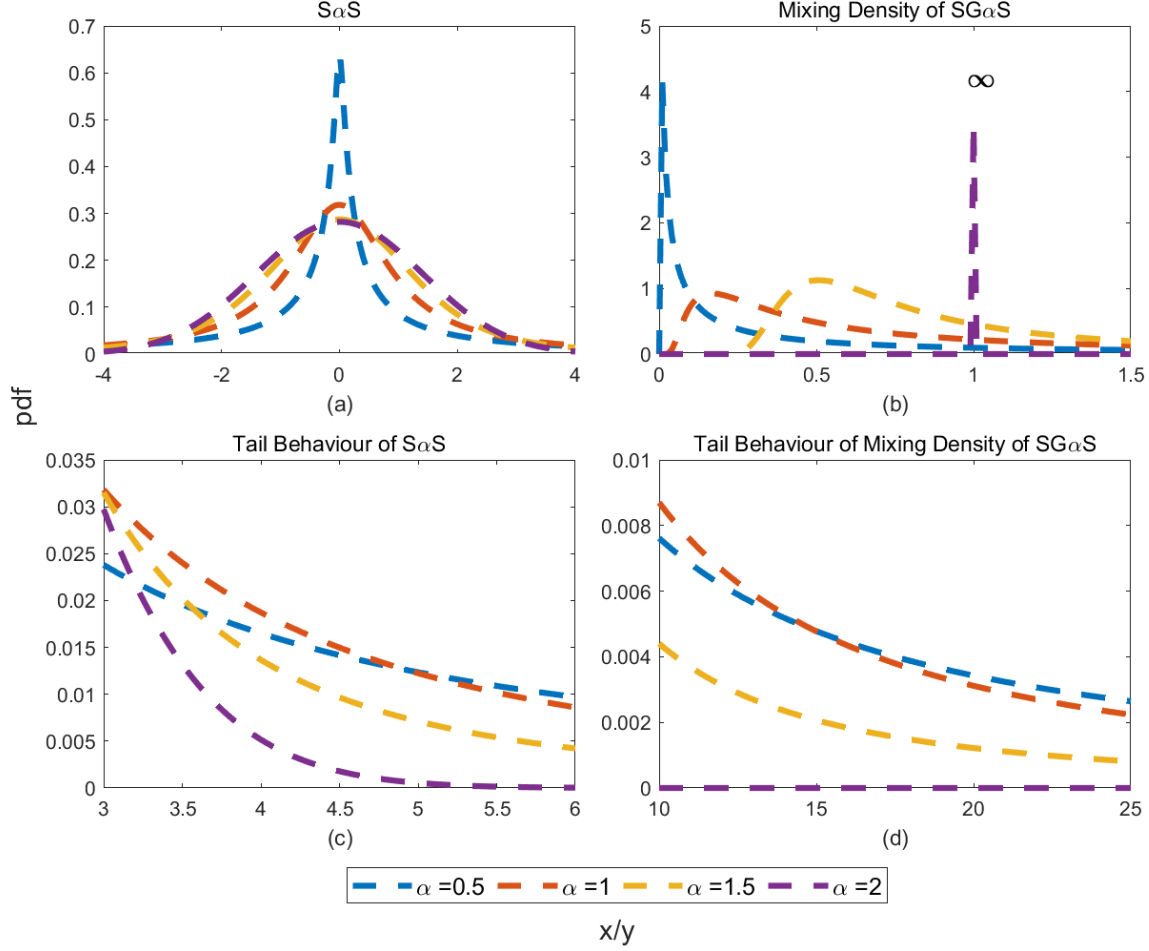


Figure 4.1: Mixing density and tail behaviour analysis. (a) and (b) plot the PDFs of the SG α S distribution and its mixing density, respectively. Also, the corresponding tail behaviour is shown in (c) and (d).

Rao–Blackwellized particle filter is employed. However, these time-series inference methods rely on computationally expensive sampling methods, such as the IS technique. This section proposes a more efficient robust Kalman filter based on the SG α S distribution. Section 4.2.1 introduces the proposed framework, and Section 4.2.2 details the four MMSE estimators used to estimate the scale function. Moreover, Section 4.2.3 describes the convergence test strategy employed for the RKF-SG α S.

4.2.1 Proposed Robust Kalman Filtering Framework

A linear state-space model is introduced in Section 1.3. Assume the zero-mean signal and measurement noises are subject to Gaussian and SG α S distributions, respectively.

Then, both are modelled as

$$(4.3) \quad p(\mathbf{w}_{k-1}) = \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$$

$$(4.4) \quad p(\mathbf{v}_k) = \int_0^{+\infty} \mathcal{N}(\mathbf{v}_k; \mathbf{0}, \lambda_k \mathbf{R}_k) S(\lambda_k) d\lambda_k$$

where \mathbf{Q}_{k-1} is the covariance matrix of the state noise at time $k-1$ whilst \mathbf{R}_k and λ_k refer to the scale matrix and mixing parameter of the measurement noise at time k , respectively. Also, the mixing density $S(\lambda_k; \alpha/2, 1, \cos(\pi\alpha/2)^{2/\alpha}, 0)$ is simplified as $S(\lambda_k)$. Hence, the forecast PDF $p(\mathbf{x}_k | \mathbf{z}_{1:k-1})$ and likelihood PDF $p(\mathbf{z}_k | \mathbf{x}_k)$ can be expressed as

$$(4.5) \quad p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \mathcal{N}(\mathbf{x}_k; \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k|k-1})$$

$$(4.6) \quad p(\mathbf{z}_k | \mathbf{x}_k) = \int_0^{+\infty} \mathcal{N}(\mathbf{z}_k; \mathbf{H}_k \mathbf{x}_k, \lambda_k \mathbf{R}_k) S(\lambda_k) d\lambda_k$$

where $\hat{\mathbf{x}}_{k-1|k-1}$ is the posterior mean vector at time $k-1$ and

$$(4.7) \quad \mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_{k-1}$$

denotes the error covariance matrix at time $k-1$. According to (4.6), $p(\mathbf{z}_k | \mathbf{x}_k)$ follows a hierarchical Gaussian form of

$$(4.8) \quad p(\mathbf{z}_k | \mathbf{x}_k, \lambda_k, \mathbf{R}_k) = \mathcal{N}(\mathbf{z}_k; \mathbf{H}_k \mathbf{x}_k, \lambda_k \mathbf{R}_k)$$

where $\lambda_k \sim S(\lambda_k)$. Assume the true value of \mathbf{R}_k is unknown, and we model its uncertainty using the inverse-Wishart (IW) distribution, i.e. $p(\mathbf{R}_k) = \text{IW}(\mathbf{R}_k; u_k, \mathbf{U}_k)$, where u_k and \mathbf{U}_k are the dof parameter and scale matrix, respectively. According to equations (4.3)-(4.8), the joint posterior distribution $p(\tilde{\Theta} | \mathbf{z}_{1:k})$ can be expressed as

$$(4.9) \quad \begin{aligned} p(\tilde{\Theta} | \mathbf{z}_{1:k}) &\propto p(\mathbf{z}_k | \tilde{\Theta}) p(\tilde{\Theta} | \mathbf{z}_{1:k-1}) p(\mathbf{z}_{1:k-1}) \\ &= \mathcal{N}(\mathbf{z}_k; \mathbf{H}_k \mathbf{x}_k, \lambda_k \mathbf{R}_k) \\ &\times \mathcal{N}(\mathbf{x}_k; \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k|k-1}) \times S(\lambda_k) \\ &\times \text{IW}(\mathbf{R}_k; u_k, \mathbf{U}_k) \times p(\mathbf{z}_{1:k-1}) \end{aligned}$$

where $\tilde{\Theta}_k = \{\mathbf{x}_k, \lambda_k, \mathbf{R}_k\}$. As there is no analytic expression for $p(\tilde{\Theta}_k | \mathbf{z}_{1:k})$, the VB approach [95] based on the EM algorithm is employed to approximate the posterior density from

$$(4.10) \quad p(\tilde{\Theta}_k | \mathbf{z}_{1:k}) \approx q(\mathbf{x}_k) q(\lambda_k) q(\mathbf{R}_k).$$

Then, the density $q(\cdot)$ can be calculated by

$$(4.11) \quad \log q(\phi) = \mathbb{E}_{\tilde{\Theta}_k^{(-\phi)}} [\log p(\tilde{\Theta}_k, \mathbf{z}_{1:k})] + c_\phi$$

where $\phi \subset \tilde{\Theta}_k$, $\phi \cup \tilde{\Theta}_k^{(-\phi)} = \tilde{\Theta}_k$ and c_ϕ is a constant number related to ϕ .

Next, the fixed-point iteration is employed to solve equation (4.11). At the $t + 1$ -th iteration, $q^{(t+1)}(\phi)$ is approximated by employing $q^{(t)}(\phi)$ to calculate the expectation in (4.11). Proposition 4.1-4.3 present the updated formulation, but their proofs are not provided in this work as similar proofs have been made in previous work [41, 104, 108].

Proposition 4.1 ([104] Proposition 2). *Let $\phi = \mathbf{x}_k$, then $q^{(t)}(\mathbf{x}_k)$ can be updated as the Gaussian distribution, i.e.,*

$$q^{(t+1)}(\mathbf{x}_k) = \mathcal{N}\left(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^{(t+1)}, \mathbf{P}_{k|k}^{(t+1)}\right),$$

where

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} \\ \mathbf{K}_k^{(t+1)} &= \mathbf{P}_{k|k-1} \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_{k|k-1}^T \mathbf{H}_k^T + \tilde{\mathbf{R}}_k^{(t)} \right)^{-1} \\ \hat{\mathbf{x}}_{k|k}^{(t+1)} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k^{(t+1)} (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}) \\ \mathbf{P}_{k|k}^{(t+1)} &= \left(\mathbf{I}_n - \mathbf{K}_k^{(t+1)} \mathbf{H}_k \right) \mathbf{P}_{k|k-1}. \end{aligned} \quad (4.12)$$

where $\tilde{\mathbf{R}}_k^{(t)}$ is the modified measurement noise covariance matrix and can be written as

$$\tilde{\mathbf{R}}_k^{(t)} = \frac{[\mathbf{E}^{(t)}(\mathbf{R}_k^{-1})]^{-1}}{\mathbf{E}^{(t)}(\kappa(\lambda_k))} = \frac{[\mathbf{E}^{(t)}(\mathbf{R}_k^{-1})]^{-1}}{\mathbf{E}^{(t)}(\lambda_k^{-1})}. \quad (4.13)$$

Proposition 4.2 ([104] Proposition 4). *Let $\phi = \lambda_k$, $\log q^{(t+1)}(\lambda_k)$ can be expressed as,*

$$\log q^{(t+1)}(\lambda_k) = -\frac{\eta}{2} \lambda_k^{-1} - \frac{m}{2} \log \lambda_k + \log S(\lambda_k) + C', \quad (4.14)$$

where C' is a constant number and

$$\begin{aligned} \eta &= \text{tr} \left\{ \mathbf{B}_k^{(t+1)} \mathbf{E}^{(t)}(\mathbf{R}_k^{-1}) \right\} \\ \mathbf{B}_k^{(t+1)} &= \left(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k}^{(t+1)} \right) \left(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k}^{(t+1)} \right)^T + \mathbf{H}_k \mathbf{P}_{k|k}^{(t+1)} \mathbf{H}_k^T \end{aligned} \quad (4.15)$$

where $\text{tr}()$ denotes trace operation.

Proposition 4.3 ([108] Proposition 4). *Let $\phi = \mathbf{R}_k$, $q^{(t+1)}(\mathbf{R}_k)$ is written as the IW distribution, i.e.,*

$$q^{(t+1)}(\mathbf{R}_k) = \text{IW}\left(\mathbf{R}_k; u_k^{(t+1)}, \mathbf{U}_k^{(t+1)}\right),$$

where

$$\begin{aligned}
 (4.16) \quad & u_k^{(t+1)} = u_k + 1, \mathbf{U}_k^{(t+1)} = \mathbf{U}_k + \mathbf{D}_k^{(t+1)} \\
 & \mathbf{D}_k^{(t+1)} = \mathbf{E}^{(t+1)}(\lambda_k^{-1}) \times \left[\mathbf{b}_k^{(t+1)} \left(\mathbf{b}_k^{(t+1)} \right)^T + \mathbf{H}_k \hat{\mathbf{P}}_{k|k}^{(t+1)} \mathbf{H}_k^T \right] \\
 & \mathbf{b}_k^{(t+1)} = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k}^{(t+1)}
 \end{aligned}$$

Thence, we have

$$(4.17) \quad \mathbf{E}^{(t+1)}(\mathbf{R}_k^{-1}) = \left(u_k^{(t+1)} - m - 1 \right) \left(\mathbf{U}_k^{(t+1)} \right)^{-1}$$

Remark 1. In equations (4.13) and (4.16), the expectation of the scale function $\mathbf{E}(\kappa(\lambda_k)) = \mathbf{E}(\lambda_k^{-1})$ is required. While Proposition 4.2 provides the PDF of λ_k , the calculation of the expectation is not straightforward due to the lack of a closed-form expression of $S(\lambda_k)$ in equation (4.14).

4.2.2 MMSE Estimators of the Scale Function

To estimate the expectation of the scale function, this section introduces four estimators of $\mathbf{E}^{(t+1)}(\lambda_k^{-1})$, including IS and GLQ-based estimators, along with their corresponding hybrid estimators with the incorporation of the GS based estimator. For simplicity, we replace λ_k with y and omit the iteration number $t + 1$ in (4.14), and then we have

$$(4.18) \quad \mathbf{E}^{(t+1)}(\lambda_k^{-1}) = \mathbf{E}(y^{-1})$$

$$(4.19) \quad \log q(y) = -\frac{\eta}{2} y^{-1} - \frac{m}{2} \log y + \log S(y) + C'$$

$$(4.20) \quad q(y) \propto q'(y) = y^{-\frac{m}{2}} \exp\left(-\frac{\eta}{2y}\right) S(y),$$

where $q'(y)$ is a function proportional to the density $q(y)$. Then,

$$(4.21) \quad \mathbf{E}(y^{-1}) = \frac{\int_0^{+\infty} y^{-1} q'(y) dy}{\int_0^{+\infty} q'(y) dy}.$$

4.2.2.1 Importance Sampling Based Estimator

Here we introduce the estimation method of $\mathbf{E}(y^{-1})$ based on the IS technique. The IS-based estimator is elaborated in Theorem 4.1.

Theorem 4.1. Let y_i ($i = 1, \dots, N$) be samples from the target distribution $S(y)$. Then $\mathbf{E}(y^{-1})$ can be estimated by

$$(4.22) \quad \mathbf{E}(y^{-1}) = \sum_{i=1}^N w_i y_i^{-1}$$

where

$$(4.23) \quad w_i = \frac{y_i^{-\frac{m}{2}} \exp\left(-\frac{\eta}{2y_i}\right)}{\sum_{i'=1}^N y_{i'}^{-\frac{m}{2}} \exp\left(-\frac{\eta}{2y_{i'}}\right)}.$$

Proof 1. According to (4.21), to calculate $E(y^{-1})$, two integrals, $\int_0^{+\infty} y^{-1} q'(y) dy$ and $\int_0^{+\infty} q'(y) dy$, need to be approximated.

Based on the IS method and (4.20), these two integrals can be approximated based on the samples y_i from $S(y)$. We rewrite

$$\begin{aligned} \int_0^{+\infty} y^{-1} q'(y) dy &\approx \frac{1}{N} \sum_{i=1}^N y_i^{-\frac{m}{2}-1} \exp\left(-\frac{\eta}{2y_i}\right) \\ \int_0^{+\infty} q'(y) dy &\approx \frac{1}{N} \sum_{i=1}^N y_i^{-\frac{m}{2}} \exp\left(-\frac{\eta}{2y_i}\right). \end{aligned}$$

Then $E(y^{-1})$ in (4.21) can be obtained by using the expressions above and Theorem 4.1 holds \square .

Remark 2. Although the approximation of the $S(y)$ value in (4.20) is computationally expensive [68], it can be efficiently sampled [102]. Hence, based on the IS algorithm, we first obtain the samples of $S(y)$, and then the numerator and denominator integrals in (4.21) can be approximated.

4.2.2.2 Gauss–Laguerre Quadrature Based Estimator

An alternative estimator of $E(y^{-1})$, this time based on the GLQ, is discussed in this section. The details of this estimator are provided in Theorem 4.2.

Theorem 4.2. Based on the GLQ method, $E(y^{-1})$ can be approximated by

$$(4.24) \quad E(y^{-1}) = \frac{\sum_{l=1}^L w_l x_l f(x_l)}{\frac{\eta}{2} \sum_{l=1}^L w_l f(x_l)}.$$

where x_l , $l = 1, \dots, L$, is the l -th root of Laguerre polynomial $\mathcal{L}_L(x)$ and $f(x_l) = x_l^{\frac{m}{2}-2} S(\frac{\eta}{2x_l})$. Also, the weight w_l can be written as

$$(4.25) \quad w_l = \frac{x_l}{(L+1)^2 [\mathcal{L}_{L+1}(x_l)]^2}.$$

Proof 2. In order to approximate the integrals in (4.21), first let $y = \frac{\eta}{2x}$, then

$$\begin{aligned}\int_0^{+\infty} q'(y)dy &= \left(\frac{\eta}{2}\right)^{-\frac{m}{2}+1} \times \int_0^{+\infty} \exp(-x)x^{\frac{m}{2}-2}S\left(\frac{\eta}{2x}\right)dx \\ \int_0^{+\infty} y^{-1}q'(y)dy &= \left(\frac{\eta}{2}\right)^{-\frac{m}{2}} \times \int_0^{+\infty} \exp(-x)x^{\frac{m}{2}-1}S\left(\frac{\eta}{2x}\right)dx\end{aligned}$$

Recalling (4.21) and replacing the expressions above in (4.21), we get

$$\mathbb{E}(y^{-1}) = \frac{\int_0^{+\infty} x \exp(-x) f(x) dx}{\frac{\eta}{2} \int_0^{+\infty} \exp(-x) f(x) dx}$$

where $f(x) = x^{\frac{m}{2}-2}S(\frac{\eta}{2x})$. Then the GLQ method in Theorem 4.2 holds \square .

Remark 3. The GLQ-based estimator in Theorem 4.2 requires calculating $f(x_l)$ in which the approximation of $S(\frac{\eta}{2x_l})$ cannot be avoided. However, compared to the IS method, the GLQ requires fewer samples [54]. Thus, despite the large computational load of approximating $S(y)$ values [68], the GLQ is still applicable to the estimation of $\mathbb{E}(y^{-1})$.

4.2.2.3 Gamma Series Based Estimator

Both the IS and GLQ-based estimators suffer from low efficiency due to their requirements of numerous particles and the approximation of $S(y)$, respectively. In contrast, a more efficient estimator based on the GS is explained in detail in this section. We first show how the density $q'(y)$ can be represented by a series based on the inverse Gamma density (cf. Lemma 4.1). Subsequently, a corresponding estimator of $\mathbb{E}(y^{-1})$ is introduced as a result of Theorem 4.3.

Lemma 4.1. For the SGaS distribution with shape parameter α , let $\alpha_1 = \frac{\alpha}{2}$, then $q'(y)$ can be represented as an Inverse-Gamma-PDF-based series, i.e.,

$$(4.26) \quad q'(y) = \sum_{\xi=1}^{+\infty} c_{\xi} \frac{\Gamma(a_{\xi})}{b^{a_{\xi}}} \text{IG}(y; a_{\xi}, b)$$

where $\Gamma(\cdot)$ is gamma function and

$$(4.27) \quad a_{\xi} = \xi \alpha_1 + \frac{m}{2}, \quad b = \frac{\eta}{2}$$

$$(4.28) \quad c_{\xi} = (-1)^{\xi+1} \frac{\Gamma(\xi \alpha_1 + 1)}{\pi \xi!} \sin(\xi \alpha_1 \pi).$$

Proof 3. For the SGaS with shape parameter α , let $\alpha_1 = \frac{\alpha}{2}$, then $S(y)$ can be represented as a series [10]

$$S(y) = -\frac{1}{\pi y} \sum_{\xi=1}^{+\infty} \frac{\Gamma(\xi \alpha_1 + 1)}{\xi!} (-y^{-\alpha_1})^{\xi} \sin(\xi \alpha_1 \pi)$$

Then, replacing $S(y)$ with the expression above in (4.20),

$$(4.29) \quad q'(y) = \sum_{\xi=1}^{+\infty} c_{\xi} y^{-a_{\xi}-1} \exp\left(-\frac{b}{y}\right)$$

where a_{ξ} , b and c_{ξ} are defined in (4.27) and (4.28). As $y^{-a_{\xi}-1} \exp\left(-\frac{b}{y}\right)$ is proportional to an IG distribution, equation (4.29) can be reformulated as in (4.26) which finalises the proof of Lemma 4.1 \square .

Theorem 4.3. For the SGαS distribution with shape parameter α , $E(y^{-1})$ can be represented as the ratio of two Gamma function-based series sums as

$$(4.30) \quad E(y^{-1}) = \frac{\sum_{\xi=1}^{+\infty} c_{\xi} \frac{\Gamma(a_{\xi}+1)}{b^{(a_{\xi}+1)}}}{\sum_{\xi=1}^{+\infty} c_{\xi} \frac{\Gamma(a_{\xi})}{b^{a_{\xi}}}}$$

iff both the series sums are convergent.

Proof 4. Based on equation (4.26), we can represent the two integrals in (4.21) with two series respectively as

$$(4.31) \quad \int_0^{+\infty} q'(y) dy = \sum_{\xi=1}^{+\infty} c_{\xi} \frac{\Gamma(a_{\xi})}{b^{a_{\xi}}} \int_0^{+\infty} \text{IG}(y; a_{\xi}, b) dy$$

$$(4.32) \quad \int_0^{+\infty} y^{-1} q'(y) dy = \sum_{\xi=1}^{+\infty} c_{\xi} \frac{\Gamma(a_{\xi})}{b^{a_{\xi}}} \int_0^{+\infty} \frac{\text{IG}(y; a_{\xi}, b)}{y} dy$$

assuming the above series are convergent. As $\int_0^{+\infty} \text{IG}(y; a_{\xi}, b) dy = 1$, from (4.31), we have

$$(4.33) \quad \int_0^{+\infty} q'(y) dy = \sum_{\xi=1}^{+\infty} c_{\xi} \frac{\Gamma(a_{\xi})}{b^{a_{\xi}}}.$$

Following this, since $y \sim \text{IG}(y; a_{\xi}, b)$, $y^{-1} \sim G(y; a_{\xi}, b)$, then

$$(4.34) \quad \int_0^{+\infty} \frac{\text{IG}(y; a_{\xi}, b)}{y} dy = \int_0^{+\infty} y G(y; a_{\xi}, b) dy = \frac{a_{\xi}}{b}.$$

According to (4.32) and (4.34), we write

$$(4.35) \quad \int_0^{+\infty} y^{-1} q'(y) dy = \sum_{\xi=1}^{+\infty} c_{\xi} \frac{\Gamma(a_{\xi}+1)}{b^{a_{\xi}+1}}$$

Using the expressions in (4.33) and (4.35), we obtain the equation (4.30) \square .

Convergence analysis: According to the ratio test of an infinite series [88], the absolute convergence conditions of the numerator and denominator in (4.30) can be written as

$$(4.36) \quad \lim_{\xi \rightarrow \infty} \left| \frac{c_{\xi+1} \frac{\Gamma(a_{\xi+1}+1)}{b^{(a_{\xi+1}+1)}}}{c_{\xi} \frac{\Gamma(a_{\xi}+1)}{b^{(a_{\xi}+1)}}} \right| < 1 \quad \text{and} \quad \lim_{\xi \rightarrow \infty} \left| \frac{c_{\xi+1} \frac{\Gamma(a_{\xi+1})}{b^{a_{\xi+1}}}}{c_{\xi} \frac{\Gamma(a_{\xi})}{b^{a_{\xi}}}} \right| < 1,$$

respectively. Then the absolute convergence condition can be simplified as

$$(4.37) \quad \lim_{\xi \rightarrow \infty} 2 \left| \frac{\Gamma(1 - \xi \alpha_1) \Gamma(a_{\xi} + \alpha_1 + 1)}{\xi \Gamma(1 - \xi \alpha_1 - \alpha_1) \Gamma(a_{\xi} + 1)} \right|^{\frac{1}{\alpha_1}} < \eta.$$

While a closed-form expression for the limit in (4.37) remains elusive, the behaviour of the series in (4.30) becomes apparent, indicating divergence as η approaches zero.

Remark 4. *The numerator and denominator series in equation (4.30) may not always converge. Hence, the estimator cannot be directly applied to the RKF-SGaS, as it fails in cases when the estimator diverges.*

4.2.2.4 Hybrid Estimators

Owing to the divergence of the GS-based estimator, two hybrid estimators, GSIS and SGaS, are further developed. When either/both the numerator or/and denominator series in (4.30) diverges, the GS estimator is replaced with the IS or GLQ-based methods.

Considering the convergence ranges of the above-mentioned series are not available, convergence analysis requires testing. For this purpose, let

$$(4.38) \quad r_{\xi}^{(1)} = c_{\xi} \frac{\Gamma(a_{\xi} + 1)}{b^{(a_{\xi}+1)}}, \quad r_{\xi}^{(2)} = c_{\xi} \frac{\Gamma(a_{\xi})}{b^{a_{\xi}}}$$

where $r_{\xi}^{(1)}$ and $r_{\xi}^{(2)}$ are the ξ -th elements of the numerator and denominator series, respectively. Then, the convergence conditions of these two series can be written as [86]

$$(4.39) \quad \sum_{\xi=\bar{\xi}-\tau_1}^{\bar{\xi}} \left| \frac{r_{\xi}^{(j)}}{\sum_{\xi'=1}^{\xi} r_{\xi'}^{(j)}} \right| < \varepsilon_1, \quad j = 1, 2$$

where the threshold ε_1 is a small positive number. $\tau_1 \geq 1$ is a small positive integer used to ensure the series sums are stable when $\bar{\xi} - \tau_1 \leq \xi \leq \bar{\xi}$. Also, we set the largest value of $\bar{\xi}$ as Ξ . Following this, we can judge the convergence with the increase of $\bar{\xi}$. If $\bar{\xi} < \Xi$ and the condition in (4.39) are satisfied, the estimation results are reliable. By contrast, we consider the estimator to diverge when $\bar{\xi} = \Xi$ and the convergence inequalities are violated. Then, either the IS or GLQ estimator is employed instead. The whole estimation method is shown in Algorithm 11.

Algorithm 11: GSIS and GSGL estimation

Input: η, m, α
 $b \leftarrow \frac{\eta}{2};$
for $\bar{\xi} \leftarrow 1$ **to** Ξ **do**
 Calculate $a_{\bar{\xi}}$ by (4.27);
 Calculate $c_{\bar{\xi}}$ by (4.28);
 Calculate $r_{\bar{\xi}}^{(1)}, r_{\bar{\xi}}^{(2)}$ by (4.38);
 if $\bar{\xi} < \Xi$ **and inequalities in (4.39) are satisfied** **then**
 Estimate $E(y^{-1})$ by (4.30); Break;
 if $\bar{\xi} = \Xi$ **then**
 Estimate $E(y^{-1})$ by (4.22) or (4.24);
return $E(y^{-1})$

4.2.3 Convergence Test for Fixed-point Iteration Method

Under the RKF-SGaS framework, the convergence of the fixed-point iteration to solve equation (4.11) needs to be investigated. In previous work [42], $\mathbf{x}_{k|k}^{(t)}$ was solely tested. However, in this chapter, we extend this approach by additionally detecting the convergence of the posterior covariance matrix $\mathbf{P}_{k|k}^{(t)}$ and the expectation of the scale function $E^{(t)}(\kappa(\lambda_k))$ to improve the detection reliability. Furthermore, we test the convergence of the latest $\tau_2 > 1$ items, as opposed to only the t -th item tested in [42]. The convergence conditions are expressed as

$$(4.40) \quad \sum_{t=\bar{t}-\tau_2}^{\bar{t}} \frac{\text{sum}(\text{abs}(\Phi^{(t)} - \Phi^{(t-1)}))}{\text{sum}(\text{abs}(\Phi^{(t)}))} < \varepsilon_2$$

where $\text{abs}(\mathbf{X})$ is element-wise absolute value operation and $\text{sum}(\mathbf{X})$ denotes summation for all the elements. Also, $\Phi \in \{\mathbf{x}_{k|k}, \mathbf{P}_{k|k}, E(\kappa(\lambda_k))\}$ and ε_2 is a small positive number. In addition, the fixed-point iteration has a maximum number of iterations, denoted by M , after which the algorithm will terminate and output the current estimation results. The implementation pseudo-code for the proposed RKF-SGaS is given in Algorithm 12.

4.3 Improved RKF-SL

In this section, we give the MMSE estimator of the scale function under the RKF-SL framework. As a special case of the GSM distribution, it has been applied to the RKF in the previous work [7, 41], whilst only the MAP estimate of the scale function was

Algorithm 12: One Time Step of the Proposed RKF-SG α S

Input: $\mathbf{z}_k, \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}, \mathbf{F}_k, \mathbf{H}_k, \mathbf{Q}_k, u_k, \mathbf{U}_k, \alpha, m, \Xi, \varepsilon_1, \varepsilon_2, \tau_1, \tau_2, M, N, L$
 Calculate $\mathbf{P}_{k|k-1}$ using (4.7);
 Initialisation: $\mathbf{E}^{(0)}(\mathbf{R}_k^{-1}) = (u_k - m - 1)(\mathbf{U}_k)^{-1}$, $\mathbf{E}^{(0)}(\lambda_k^{-1}) = 1$;
for $t \leftarrow 0$ **to** $M - 1$ **do**
 Calculate $\tilde{\mathbf{R}}_k^{(t)}$ using (4.13);
 Calculate $\hat{\mathbf{x}}_{k|k}^{(t+1)}$ and $\mathbf{P}_{k|k}^{(t+1)}$ using (4.12);
 Calculate $\mathbf{B}_k^{(t+1)}$ and η using (4.15);
 Calculate $\mathbf{E}^{(t+1)}(\lambda_k^{-1})$ using one of the scale function estimators in
 Section 4.2.2;
 Calculate $\mathbf{b}_k^{(t+1)}, \mathbf{D}_k^{(t+1)}, u_k^{(t+1)}$ and $\mathbf{U}_k^{(t+1)}$ using (4.16);
 Calculate $\mathbf{E}^{(t+1)}(\mathbf{R}_k^{-1})$ using (4.17);
 if *The convergence condition in (4.40) is satisfied* **then**
 └ Terminate the iteration;
return $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k}^{(M)}, \mathbf{P}_{k|k} = \mathbf{P}_{k|k}^{(M)}$

employed. To enhance the precision, we derived the MMSE estimate and Proposition 4.4 is presented with details.

Proposition 4.4. *For a slash distribution with dof parameter v , $\mathbf{E}(\kappa(y))$ can be estimated by,*

$$(4.41) \quad \mathbf{E}(\kappa(y)) = \mathbf{E}(y) = \frac{\tilde{\gamma}(a+1, b)}{b\tilde{\gamma}(a, b)},$$

where $\tilde{\gamma}()$ represents lower incomplete gamma function and

$$(4.42) \quad a = \frac{m+v}{2} \quad \text{and} \quad b = \frac{\eta}{2}.$$

Proof 5. *For a slash distribution, the mixing density follows $\text{Be}(y; \frac{v}{2}, 1)$ and $0 < y < 1, v > 0$. Then according to (4.19), we have*

$$q'(y) \propto y^{\frac{m+v-2}{2}} \exp\left(-\frac{\eta}{2}y\right).$$

Thus $q'(y)$ is proportional to a Gamma PDF when $0 < y < 1$. As $\kappa(y) = y$, we have

$$(4.43) \quad \mathbf{E}(\kappa(y)) = \mathbf{E}(y) = \frac{\int_0^1 y q'(y) dy}{\int_0^1 q'(y) dy}$$

Calculate a and b according to (4.42), then

$$(4.44) \quad \begin{aligned} \int_0^1 y q'(y) dy &= \frac{\tilde{\gamma}(a+1, b)}{b^{a+1}} \\ \int_0^1 q'(y) dy &= \frac{\tilde{\gamma}(a, b)}{b^a}. \end{aligned}$$

The expressions in (4.43) and (4.44), give the expression in (4.41) which finalises the proof \square .

4.4 Numerical simulations

4.4.1 Target Tracking Models and Noises

In this section, we introduce the target tracking model used in the experimental analysis. The target moves uniformly in a straight line, and the state-space model is given by (1.3)

$$(4.45) \quad \mathbf{F}_k = \begin{bmatrix} \mathbf{I}_2 & \Delta t \mathbf{I}_2 \\ \mathbf{0} & \mathbf{I}_2 \end{bmatrix}, \quad \mathbf{H}_k = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0} \end{bmatrix}$$

with the observation interval of $\Delta t = 1$. Furthermore, the process noise \mathbf{w}_k follows the Gaussian distribution, of which the covariance matrix is given by

$$(4.46) \quad \mathbf{Q}_k = 0.1 * \overline{\mathbf{Q}}, \quad \overline{\mathbf{Q}} = \begin{bmatrix} \frac{\Delta t^3}{3} \mathbf{I}_2 & \frac{\Delta t^2}{2} \mathbf{I}_2 \\ \frac{\Delta t^2}{2} \mathbf{I}_2 & \Delta t \mathbf{I}_2 \end{bmatrix}.$$

Then, three kinds of heavy-tailed measurement noises are selected—the GM, ST and SGaS noises. Among them, the GM noise has been employed to test the filtering performance of RKF's in many references [4, 41, 72]. In contrast, we also consider the ST and SGaS noises thanks to their capability to fit the practical noises. The GM noise [41, 104] is written as

$$(4.47) \quad \mathcal{GM} = \begin{cases} \mathcal{N}(\mathbf{0}, \overline{\mathbf{R}}) & \text{with probability } 0.9 \\ \mathcal{N}(\mathbf{0}, \rho \overline{\mathbf{R}}) & \text{with probability } 0.1, \end{cases}$$

where $\overline{\mathbf{R}} = 10\mathbf{I}_2$ is the nominal covariance matrix. Also, ρ is the augment factor which is taken from a vector $[5, 10, 10^2, 10^3, 10^4, 10^5, 10^6, 10^7, 10^8]$. Contrarily, the SGaS and ST noises have the same scale matrix $\overline{\mathbf{R}}$, and their shape parameters, α and ν , take values from $[0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 1.7, 1.85]$ and $[0.3, 0.5, 0.7, 0.9, 1.2, 1.7, 2.5, 3.5, 6]$, respectively. The whole tracking process lasts for 100 seconds, and 100 Monte Carlo runs are simulated.

4.4.2 Benchmark Filters

The RKF-SGaS is compared with 3 kinds of heavy-tailed-distribution-based RKF's in the following simulations. The first two filters are the RKF based on the variance Gamma

distribution (RKF-VG) [41] and the RSTKF [39]. Lastly, our improved version of RKF-SL is also used as a comparison method. Besides, the standard KF with true noise covariance matrices (KFTNCM) [104] is employed for reference.

For the RKF-SG α S, the performance of the four variants is evaluated, including RKF-SG α S-IS, RKF-SG α S-GLQ, RKF-SG α S-GSIS, RKF-SG α S-GSGL. Then, for the GS estimator, $\Xi = 30$, $\varepsilon_1 = 10^{-2}$ and $\tau_1 = 4$. Moreover, we set $M = 50$, $\varepsilon_2 = 10^{-2}$ and $\tau_2 = 4$ for all the RKF-based filters.

4.4.3 Estimation of RKF Parameters

In this section, we employ the EM or maximum likelihood estimation (MLE) method to initialise the shape parameters of the heavy-tailed distributions, u_k and \mathbf{U}_k . Compared with manually setting these RKF parameters as in the previous work [41], this can alleviate the influence of the model errors and provide a fairer comparison among these RKF. For the stable distribution, different parameter estimation methods have been presented, such as MLE [69], the moment-based method [3], Bayesian Monte Carlo inference [58]. In this simulation, the EM algorithm in [91] is employed. Specifically, the EM iteration number is 200, and the estimates of the last 15 iterations are averaged for the final estimate. Besides, to sample the posterior distribution of the Weibull random variable under this EM parameter estimation framework, a rejection sampling method is used and the sample number is 2000.

In contrast to the case of SG α S mentioned above, the MLE algorithm is utilised for the parameter estimation of the variance Gamma (VG), slash and ST distributions. As we only consider the zero-mean symmetric noise, the values of both the mean and skewness vectors for all the distributions are assumed to be known. The measurement noise sample size is 1000, and 1000 Monte Carlo runs are taken. The shape parameter (α or ν) is estimated by averaging the results. Lastly, 1000 scale matrix estimation results are modelled by the IW distribution, and the MLE method is used to estimate its parameters u_k and \mathbf{U}_k .

4.4.4 Performance Evaluation of RKF-SG α S Variants

Based on the parameter estimation results, the properties of our proposed RKF-SG α S variants are investigated in this section. The particle number N and the root number L are taken from the vectors [100, 400, 1000, 4000, 10000, 40000, 100000] and [1, 2, 6, 10,

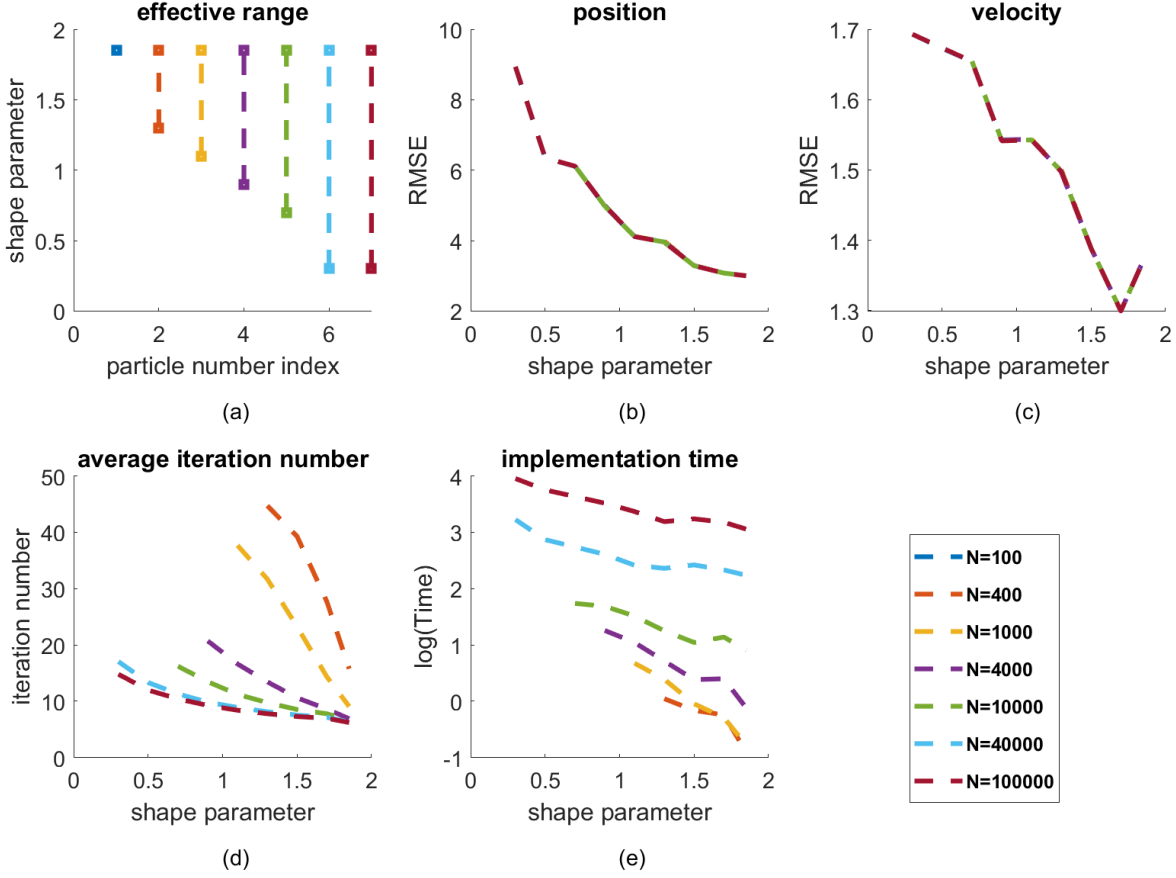


Figure 4.2: The target tracking results of the RKF-SGaS-IS under different particle numbers N . For (a), the x-axis is the particle number index and the y-axis represents the shape parameter value. In comparison, for (b)-(e), the x-axis is the shape parameter value, and the y-axis represents the estimation RMSE, iteration number and $\log(\text{time})$, respectively. (a) depicts the shape parameter ranges, where the filter can track the target steadily. By contrast, (b) and (c) show the position and velocity estimation RMSEs, respectively. Also, (d) describes the average fixed-point iteration numbers. Besides, the implementation time is shown in (e).

30, 50, 100], respectively. Also, the measurement noises come from SGaS distributions, and the parameter selection is explained in Section 4.4.1.

Figure 4.2 shows the experimental results of RKF-SGaS-IS. (1) From Figure 4.2-(a), larger particle numbers allow for the broader effective ranges of α . The heavier-tailed distributions contain larger probability spaces, and then more samples are needed for the IS approximation. (2) The position and velocity estimation RMSEs in Figure 4.2-(b,c) are not influenced by N within the effective ranges of α . This suggests that the larger estimation errors of $E(\kappa(\lambda_k))$ caused by small N have little impact on the estimation of

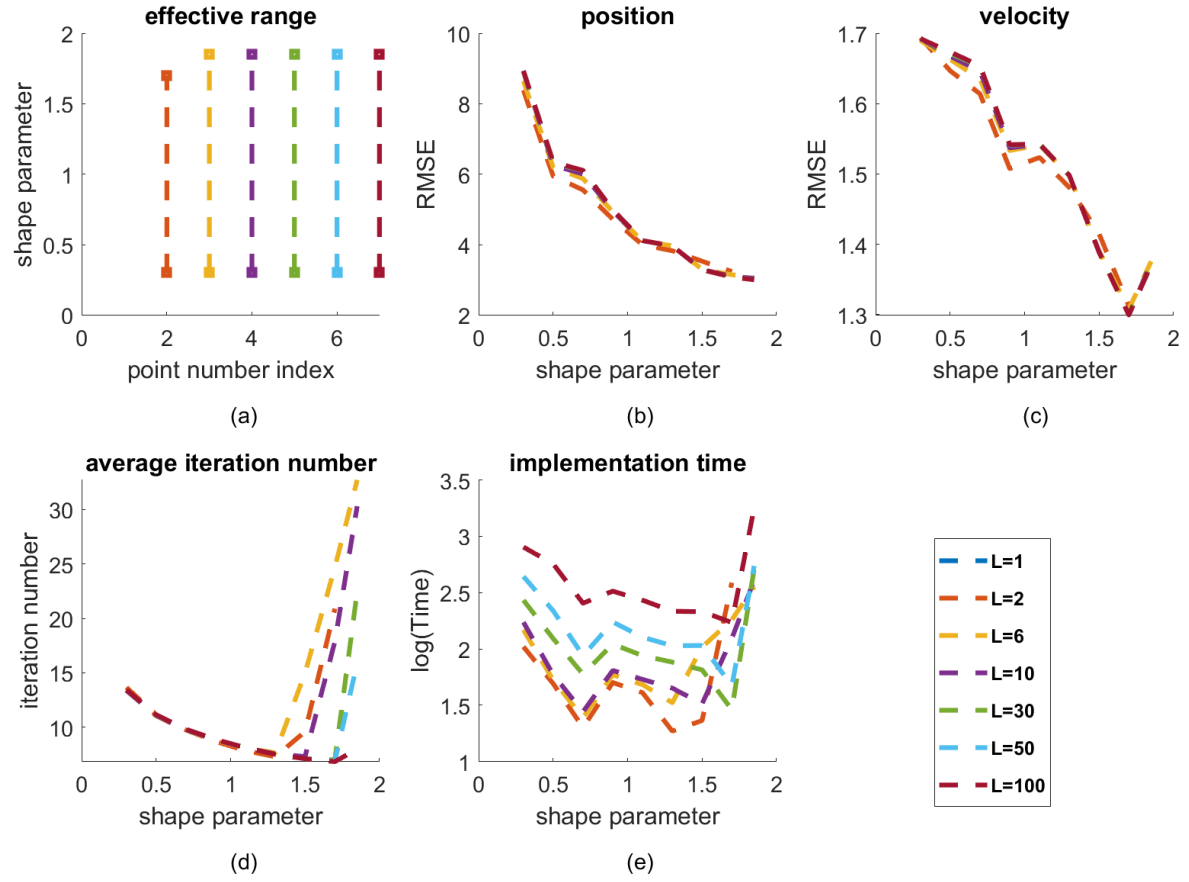


Figure 4.3: The target tracking results of the RKF-SG α S-GLQ under different root numbers L . For (a), the x-axis is the root number index and the y-axis represents the shape parameter value. In comparison, for (b)-(e), the x-axis is the shape parameter value, and the y-axis represents the estimation RMSE, iteration number and $\log(\text{time})$, respectively. (a) depicts the shape parameter ranges, where the filter can track the target steadily. By contrast, (b) and (c) show the position and velocity estimation RMSEs, respectively. Also, (d) describes the average fixed-point iteration numbers. Besides, the implementation time is shown in (e).

the position and velocity marginal distributions. (3) From Figure 4.2-(d), the average iteration number is inversely proportional to N , as the imprecise estimate of $E(\kappa(\lambda_k))$ causes the EM estimator to fluctuate around the local optimum and cannot satisfy the convergence inequalities in (4.40). Thus, the convergence process is delayed. (4) From Figure 4.2-(e), the execution time is proportional to the sample size. Although large N can reduce the iteration number, the filter with small N is still more efficient because of the shorter implementation time on the IS.

The tracking performance of RKF-SG α S-GLQ is plotted in Figure 4.3. (1) The effective

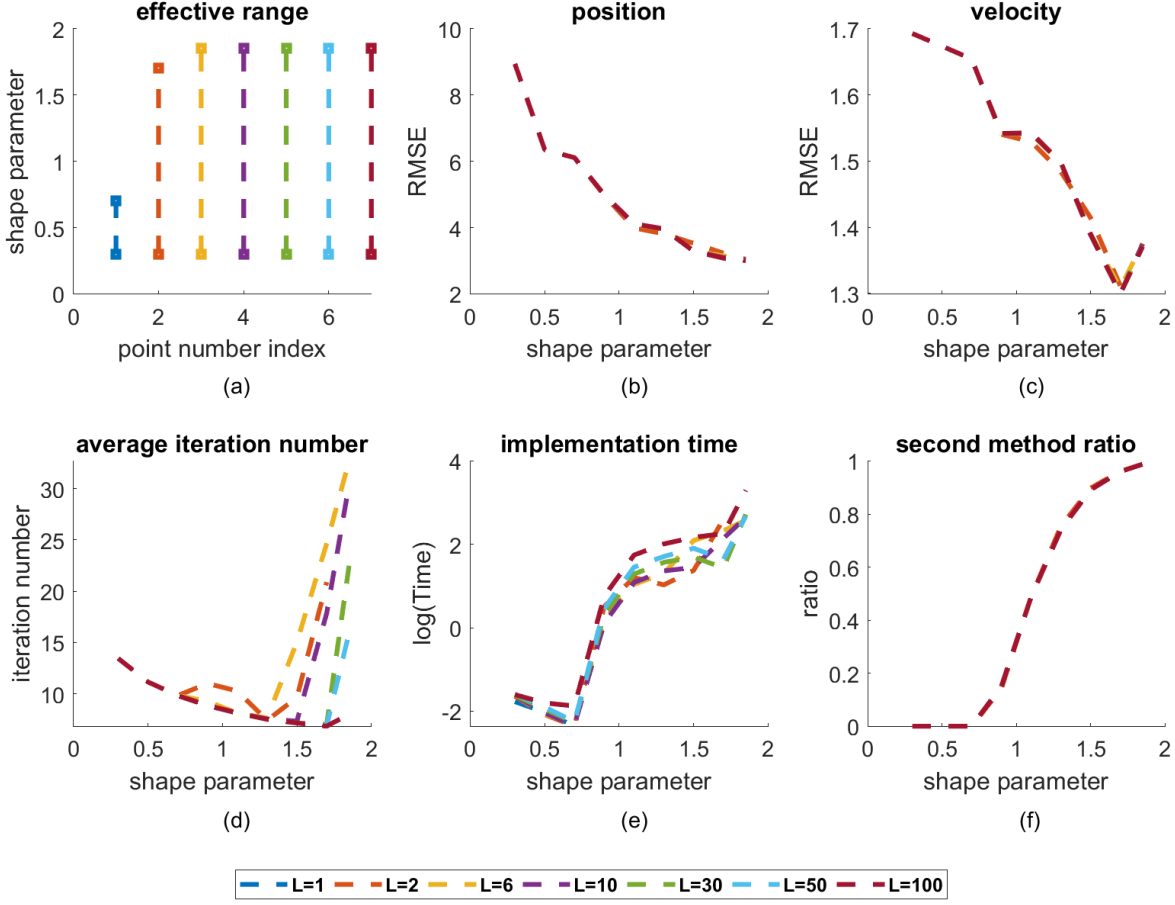


Figure 4.4: The target tracking results of the RKF-SG α S-GSGL under different root numbers L . For (a), the x-axis is the root number index and the y-axis represents the shape parameter value. In comparison, for (b)-(f), the x-axis is the shape parameter value, and the y-axis represents the estimation RMSE, iteration number, log(time) and ratio value, respectively. (a) depicts the shape parameter ranges, where the filter can track the target. By contrast, (b) and (c) shows the position and velocity estimation RMSEs, respectively. Also, (d) describes the average fixed-point iteration numbers. Then, the implementation time is shown in (e). Besides, (f) paints the ratio of the iterations where the GS method is replaced with the GLQ estimator.

α range shown in Figure 4.3-(a) is related to the root number. For $L = 1$, the filter fails on the whole α range, because the GLQ cannot estimate the integral. By contrast, for $L = 2$, the effective range is large. However, when α is large, the target is lost. For the light-tailed case, the probability space of $S(y)$ concentrates on a small range, and then the GLQ produces rough estimates. In comparison, the filter is stable on the whole tested ranges when $L \geq 3$. (2) The position and velocity estimation (Figure 4.3-(b,c)) precision

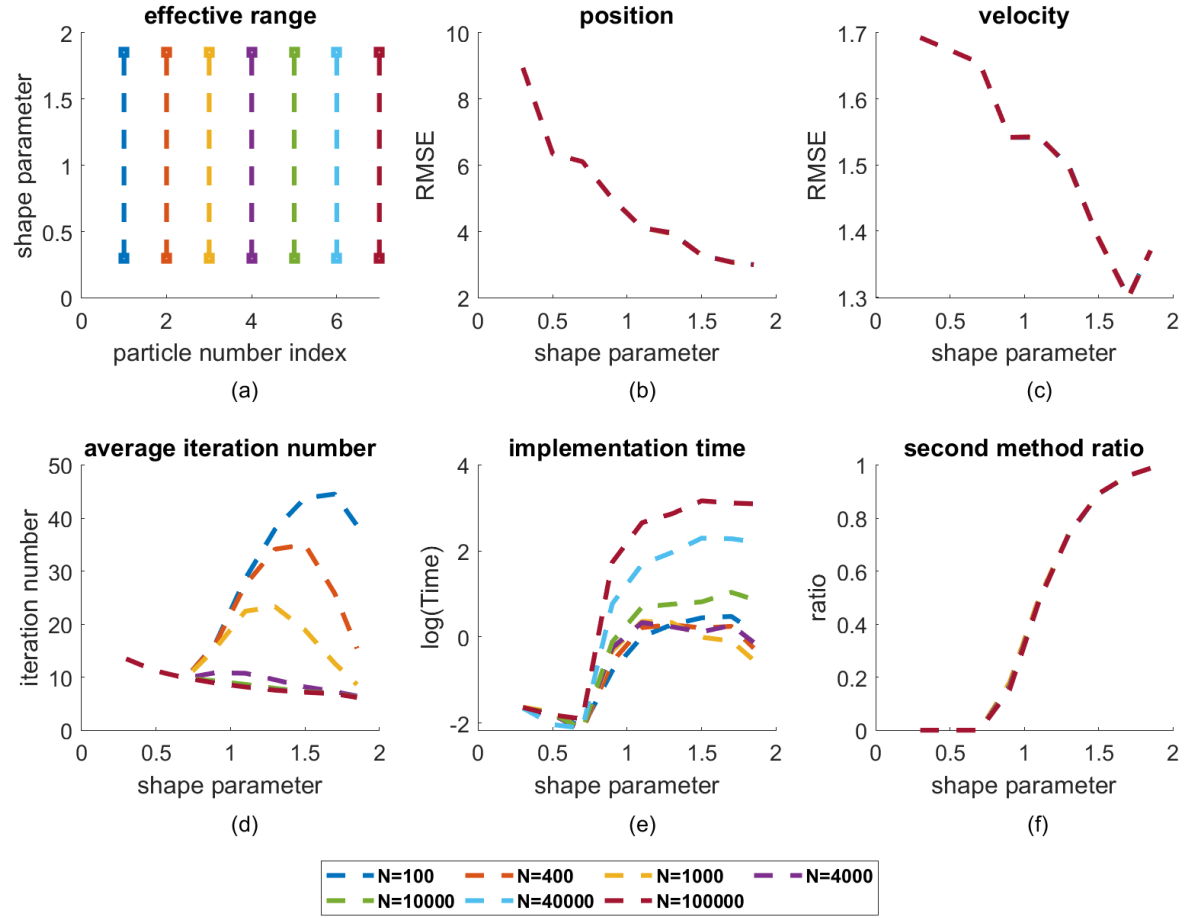


Figure 4.5: The target tracking results of the RKF-SG α S-GSIS under different particle numbers N . For (a), the x-axis is the root number index and the y-axis represents the shape parameter value. In comparison, for (b)-(f), the x-axis is the shape parameter value, and the y-axis represents the estimation RMSE, iteration number, log(time) and ratio value, respectively. (a) depicts the shape parameter ranges, where the filter can track the target. By contrast, (b) and (c) shows the position and velocity estimation RMSEs, respectively. Also, (d) describes the average fixed-point iteration numbers. Then, the implementation time is shown in (e). Besides, (f) depicts the ratio of the iterations where the GS method is replaced with the IS estimator.

cannot be improved by larger L although the GLQ can produce more precise estimates of $E(\kappa(\lambda_k))$. (3) From Figure 4.3-(d), the convergence of the fixed-point iteration method is influenced by both α and L . When α is large, smaller L requires more iterations due to the fluctuation around the local optimum. By contrast, when $\alpha < 1.3$, different values of L need similar iteration numbers since the GLQ estimation with small L is still precise in the heavy-tailed cases. (4) From Figure 4.3-(e), the relation between the execution

time and L is complicated. When α is small, the filter with small L is more efficient. Nevertheless, for large α , its efficiency degrades caused by the increased number of iterations.

The simulation results of the RKF-SG α S-GSGL are plotted in Figure 4.4. (1) For this hybrid-estimator-based filter, the ratio of the second method, GLQ, is added to Figure 4.4-(f). For small α , the ratio is 0. However, with the increase of α , the ratio of the GLQ gradually rises towards 1. This suggests that the GS estimator is more stable when the noise is heavy-tailed. (2) The effective range is related to L . For $L = 1$, although the RKF-SG α S-GLQ always fails in Figure 4.3-(a), the effective range of this hybrid filter is $\alpha \leq 0.7$, where the GS estimator always converges. By contrast, for $L > 1$, the effective range is the same as that of the RKF-SG α S-GLQ. (3) The position and velocity estimation results are always precise within the effective ranges. (4) The iteration number and the execution-time lines overlap for small α , where the hybrid filter mainly relies on the GS estimator. However, from Figure 4.3-(e) and 4.4-(e), the RKF-SG α S-GSGL is more efficient than the RKF-SG α S-GLQ in heavy-tailed cases, which is caused by the smaller computation loads of the GS estimator. By contrast, with the increase of α , their results gradually become similar due to the rising ratio of the GLQ estimator.

The performance of RKF-SG α S-GSIS is evaluated in Figure 4.5. (1) The ratio of the second method, the IS method, is similar to that of the RKF-SG α S-GSGL in Figure 4.4-(f). (2) The hybrid filter is stable on the whole α range, even when N is small. The GS estimator is unstable for large α , whilst the IS method with few particles can produce precise results in this case (cf. Figure 4.2). So, the complementary advantages of the GS and IS strategies extend the effective range of the filter. (3) The hybrid filter can work in the α ranges where both the IS and GS-based filters fail. For example, from Figures 4.2-(a) and 4.4-(a), the IS estimator with $N = 100$ and GS estimator are effective when $\alpha = 1.85$ and $\alpha \leq 0.7$, respectively. However, Figure 4.5-(a) shows the hybrid filter with $N = 100$ can also work when $0.7 < \alpha < 1.85$. This suggests that although the GS estimator cannot converge at every iteration in this middle range, the IS method is effective when the GS diverges. (4) All the position and velocity estimation RMSEs are close. (5) Based on Figures 4.2-(d-e) and 4.5-(d-e), the hybrid filter is more efficient than the RKF-SG α S-IS when α is small because of the high ratio of the GS estimator as shown in Figure 4.5-(f). By contrast, their performances are similar when α is large.

In summary, for the effective range, the performance of RKF-SG α S-IS is worse than the other filters, and numerous particles are needed for small α . Conversely, the position and velocity estimation results of all four proposed strategies are similar in the effective

α ranges. However, benefiting from the complementary advantages of the GS and IS estimators, the RKF-SG α S-GSIS with few particles performs steadily and then becomes the most efficient.

4.4.5 RKF-SG α S vs. Benchmark Filters

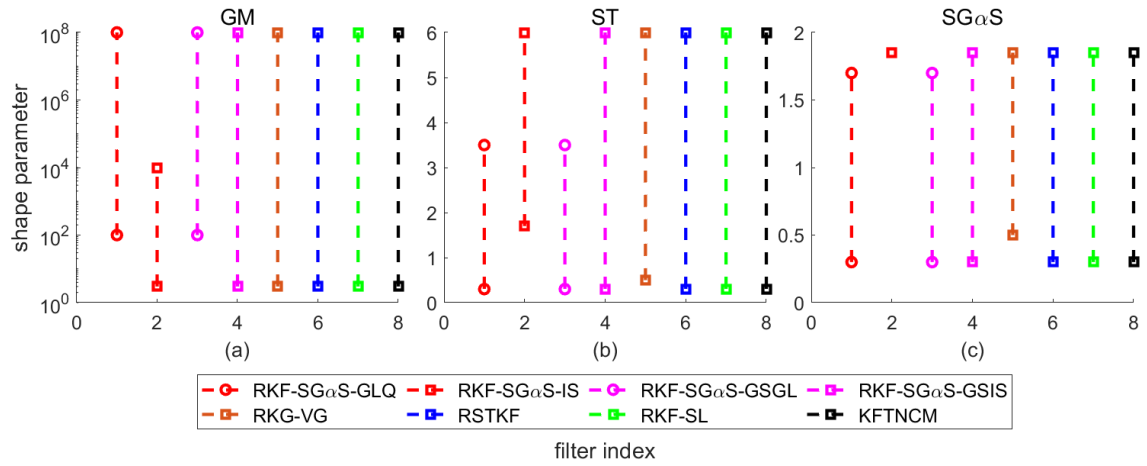


Figure 4.6: The effective shape parameter ranges of different filters under the three kinds of measurement noises. The titles of the subplots are the noise distributions. Also, for simplicity, the parameter ρ of GM distribution is defined as the shape parameter.

Following the thorough analysis of the RKF-SG α S in the previous sections, its comparison with the benchmark filters, listed in Section 4.4.2, is presented in this section. For efficiency, we set $L = 2$ and $N = 100$. Also, three kinds of measurement noises are employed, and their parameter selection is made as explained in Section 4.4.1.

In Figure 4.6, the effective range results are shown. The RKF-VG fails when the shape value is small under the ST and SG α S noises. On the contrary, all other benchmark filters behave stably over the whole tested range. However, the effective ranges of the RKF-SG α S variants are different. Particularly, RKF-SG α S-GSIS presents steady performance over the whole interval. Nevertheless, RKF-SG α S-IS fails when the tails of the noises are heavy. Conversely, in the light-tailed noise scenarios, RKF-SG α S-GLQ and RKF-SG α S-GSGL fail.

The position and velocity estimation results are depicted in Figure 4.7. Under light-tailed noise, all the filters produce similar results. However, under the heavy-tailed noises, their performances vary. Compared to RKF-VG, the RKF-SG α S always performs better under different measurement noises. Also, the RKF-SL obtains a more precise

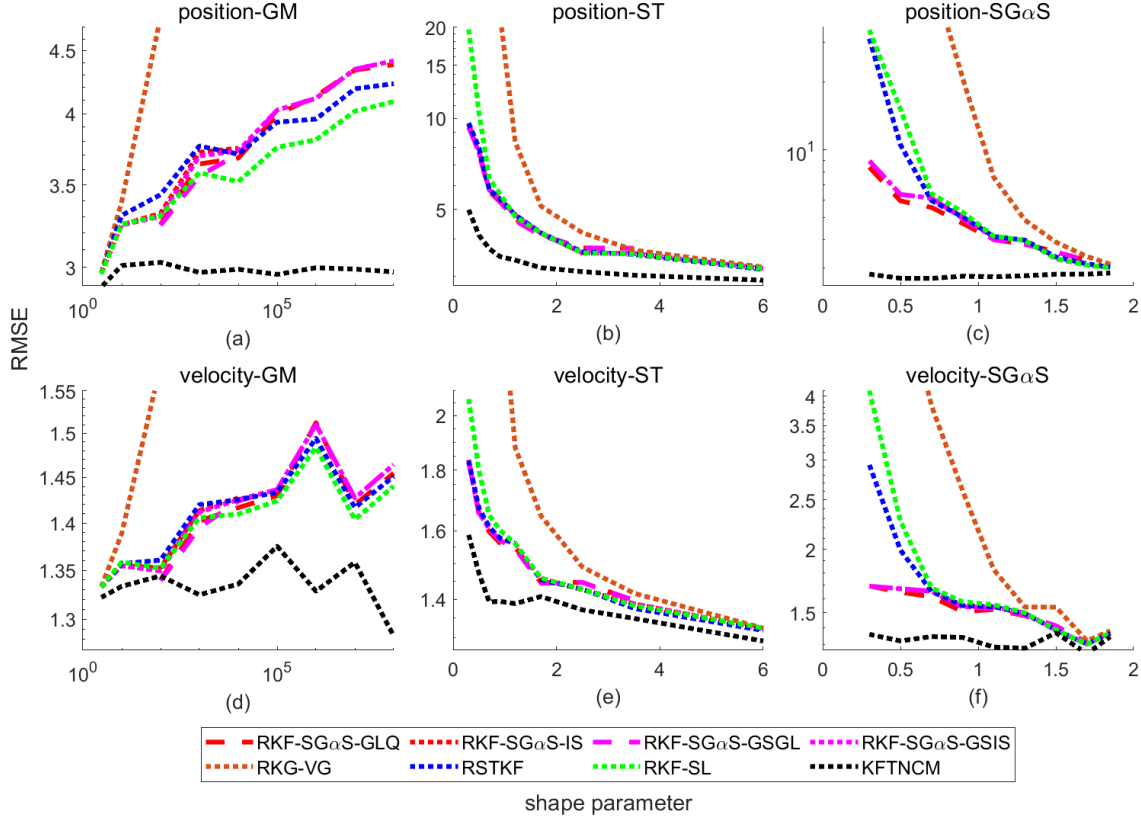


Figure 4.7: The position and velocity estimation RMSEs of different filters under the three kinds of measurement noises. The titles of the subplots follow the format: position/velocity-noise distributions. Also, for simplicity, the ρ of GM distribution is defined as the shape parameter.

estimation than RKF-SG α S under the GM noise, but worse results under ST and SG α S noises. Further, the RSTKF and RKF-SG α S achieve comparable estimates under GM and ST noises. Nonetheless, the RKF-SG α S becomes superior under the heavy-tailed SG α S noise. In addition, for different RKF-SG α S variants, the position and velocity estimation RMSEs are close in the effective ranges.

Figure 4.8 plots the execution time of different filters. Compared to RKF-SG α S-IS and RKF-SG α S-GLQ, the benchmark filters are more efficient. However, the comparison between the hybrid filters and benchmark filters is more complicated. Specifically, under the light-tailed noises, the computational loads of the hybrid filters are higher. Nevertheless, their results become close in the heavy-tailed cases because the high ratio of the GS estimator increases the filtering efficiency.

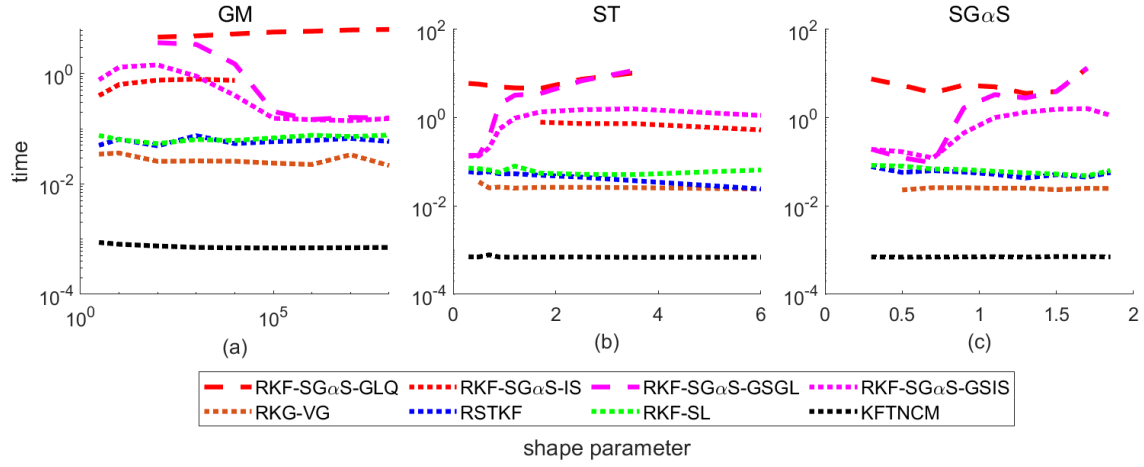


Figure 4.8: The execution time of different filters under the three kinds of measurement noises. The titles of the subplots are the noise distributions. Also, for simplicity, the ρ of GM distribution is defined as the shape parameter.

4.5 Conclusion

In this chapter, a new robust Kalman filter framework based on the $\text{SG}\alpha\text{S}$ distribution is proposed. First, we present four RKF- $\text{SG}\alpha\text{S}$ variants based on the different MMSE estimators of the scale function. Then their properties are investigated. The experimental results indicate that the RKF- $\text{SG}\alpha\text{S}$ -GSIS obtains the most efficient performance thanks to the complementary advantages of the GS and IS estimators. Second, the RKF- $\text{SG}\alpha\text{S}$ is compared with the state-of-art RKF's. From the simulations, the RKF- $\text{SG}\alpha\text{S}$ produces similar estimates under the GM and ST noises but obtains the best performance under the heavy-tailed $\text{SG}\alpha\text{S}$ noise. Considering the importance of the α -stable noise, this superiority over benchmark filters is significant. Besides, although our RKF- $\text{SG}\alpha\text{S}$ has higher computational complexity than the benchmark filters, their execution times become closer in particularly heavy-tailed cases.

In the future, we will consider applying the proposed framework to the state-space models with both heavy-tailed signal and measurement noises. In this work, only the measurement noise is considered heavy-tailed since the parameter estimation of the one-step prediction noise is more difficult compared to that of the measurement noise. However, it is straightforward to apply the RKF- $\text{SG}\alpha\text{S}$ to dynamic models with heavy-tailed state noise.

RKFNET: A NOVEL DATA-DRIVEN ROBUST KALMAN FILTER

The proposed RKF-SG α S in the last chapter presents an efficient filtering framework for linear models with heavy-tailed noise, and its robustness is proved by various systems in Section 4.4. However, its performance relies on accurate model parameters, and large model errors can degrade its performance. By contrast, in this chapter, we introduce an innovative RKF method, which combines the heavy-tailed-distribution-based RKF framework with the deep-learning technique and eliminates the need for precise parameters of heavy-tailed distributions in traditional RKF.

The rest of this chapter is organised as follows: We begin with the details of the proposed hierarchical Gaussian state-space model. Then, Section 5.2 provides the architecture of the proposed RKFnet and the USS technique is explained in Section 5.3. Furthermore, the proposed framework is tested in target tracking scenarios in Section 5.4, whilst Section 5.5 concludes this work with a summary.

5.1 New hierarchical Gaussian state-space model based on an unknown heavy-tailed distribution

In this section, we explain our proposed state-space model, which is based on a linear model with Gaussian signal noise and unknown heavy-tailed measurement noise. We assume the symmetric heavy-tailed distribution of the measurement noise can be written

in a hierarchical Gaussian form, i.e.

$$p(\mathbf{v}_k; \mathbf{R}) = \int_0^{+\infty} \mathcal{N}(\mathbf{v}_k; \mathbf{0}, \lambda_k \mathbf{R}) \pi(\lambda_k) d\lambda_k,$$

where \mathbf{R} is the unknown scale matrix and $\pi(\lambda_k)$ is the unknown mixing density. Then the likelihood PDF can be expressed as

$$(5.1) \quad p(\mathbf{z}_k | \mathbf{x}_k; \mathbf{R}) = \int_0^{+\infty} \mathcal{N}(\mathbf{z}_k; \mathbf{H}_k \mathbf{x}_k, \lambda_k \mathbf{R}) \pi(\lambda_k) d\lambda_k.$$

Besides, as we only consider the Gaussian state noise, the forecast PDF is the same as in equation (4.5).

Remark 5. *There is a radical difference between our proposed state-space model and the ones in the previous work. In the traditional RKF frameworks, various heavy-tailed distributions have been utilised to fit the measurement noise, and the corresponding likelihood PDFs can be seen as an approximation of equation (5.1). Specifically, $\pi(\lambda_k)$ is approximated by a fully skewed mixing density, and the uncertainty about the scale matrix \mathbf{R} is represented as an IW distribution [41]. Although showing robustness and efficiency in many scenarios, the RKFs may fail when the model error is large. By contrast, an unknown heavy-tailed noise distribution is employed in our new state-space model, and then the approximation error is eliminated.*

5.2 Proposed RKFnet architecture

Based on the forecast and likelihood PDFs in (5.1), the joint posterior distribution can be expressed as

$$(5.2) \quad \begin{aligned} p(\tilde{\Theta} | \mathbf{z}_{1:k}; \mathbf{R}) &\propto p(\mathbf{z}_k | \tilde{\Theta}; \mathbf{R}) p(\tilde{\Theta} | \mathbf{z}_{1:k-1}) p(\mathbf{z}_{1:k-1}) \\ &= \mathcal{N}(\mathbf{z}_k; \mathbf{H}_k \mathbf{x}_k, \lambda_k \mathbf{R}) \mathcal{N}(\mathbf{x}_k; \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k|k-1}) \pi(\lambda_k) p(\mathbf{z}_{1:k-1}). \end{aligned}$$

where $\tilde{\Theta}_k = \{\mathbf{x}_k, \lambda_k\}$, and then a new RKF framework can be derived as explained in Theorem 5.1.

Theorem 5.1. *(A similar proof can be seen in [41] Theorem 1) Given the marginal posterior distribution of λ_k ,*

$$(5.3) \quad p(\lambda_k | \mathbf{z}_{1:k}; \mathbf{R}) = \int p(\tilde{\Theta} | \mathbf{z}_{1:k}; \mathbf{R}) d\mathbf{x}_k,$$

the marginal posterior distribution of \mathbf{x}_k can be approximated as a Gaussian distribution, i.e.,

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}; \mathbf{R}) \approx \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}),$$

where

$$\begin{aligned}
 \hat{\mathbf{x}}_{k|k-1} &= \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} \\
 \mathbf{K}_k &= \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \tilde{\mathbf{R}}_k)^{-1} \\
 \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}) \\
 \mathbf{P}_{k|k} &= (\mathbf{I}_n - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}.
 \end{aligned}
 \tag{5.4}$$

while $\tilde{\mathbf{R}}_k$ is the modified measurement noise covariance matrix and can be written as

$$\tilde{\mathbf{R}}_k = \frac{1}{\mathbb{E}(\lambda_k^{-1})} \mathbf{R},
 \tag{5.5}$$

where

$$\mathbb{E}(\lambda_k^{-1}) = \int_0^{+\infty} \lambda_k^{-1} p(\lambda_k | \mathbf{z}_{1:k}, \mathbf{R}) d\lambda_k.$$

Remark 6. Theorem 5.1 provides an efficient RKF framework for the estimation of \mathbf{x}_k . However, in equation (5.5), the calculation of $\tilde{\mathbf{R}}_k$ requires the unknown \mathbf{R} and $\frac{1}{\mathbb{E}(\lambda_k^{-1})}$.

To estimate $\frac{1}{\mathbb{E}(\lambda_k^{-1})}$ and \mathbf{R}_k , and produce the posterior state estimation, we present a new neural network architecture, the RKFnet, combining the RKF framework in Theorem 5.1 with a deep learning technique. As shown in Figure 5.1, the RKFnet consists of three blocks, where Block I produces the posterior state estimation based on the KF framework in equation (5.4). Also, $\frac{1}{\mathbb{E}(\lambda_k^{-1})}$ and \mathbf{R} are estimated by Block II and Block III, respectively. The details of the RKFnet are explained below:

1) At any time k , Block I receives the new observation \mathbf{z}_k and the estimated $\tilde{\mathbf{R}}_k$, and then $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ are estimated by (5.4).

2) To estimate $\frac{1}{\mathbb{E}(\lambda_k^{-1})}$, Block II employs an FCN, and the input features are selected as $\Delta \mathbf{z}_{k|k-1} = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T$ according to Proposition 5.1. As \mathbf{R} and $\pi(\lambda_k)$ are fixed, they are not incorporated as input features.

Proposition 5.1. At any time k , given the new observation \mathbf{z}_k , the stochastic properties of $p(\lambda_k | \mathbf{z}_{1:k}, \mathbf{R})$ and $\frac{1}{\mathbb{E}(\lambda_k^{-1})}$ are determined by $\Delta \mathbf{z}_{k|k-1}$, $\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T$, \mathbf{R} and $\pi(\lambda_k)$.

Proof 6. According to equations (5.2) and (5.3), the marginal posterior distribution of λ_k can be expressed as

$$p(\lambda_k | \mathbf{z}_{1:k}, \mathbf{R}) \propto \int \mathcal{N}(\mathbf{z}_k; \mathbf{H}_k \mathbf{x}_k, \lambda_k \mathbf{R}) \mathcal{N}(\mathbf{x}_k; \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k|k-1}) \pi(\lambda_k) p(\mathbf{z}_{1:k-1}) d\mathbf{x}_k.$$

By integrating \mathbf{x}_k , then

$$p(\lambda_k | \mathbf{z}_{1:k}; \mathbf{R}) \propto \mathcal{N}(\Delta \mathbf{z}_{k|k-1}; \mathbf{0}, \mathbf{H} \mathbf{P}_{k|k-1} \mathbf{H}^T + \lambda_k \mathbf{R}) \pi(\lambda_k).$$

Hence we have Proposition 5.1. \square

Considering the outliers of the input features, $\text{sign}(\cdot) \log(1 + |\cdot|)$ is employed to shrink the values. Also, as $\mathbf{H} \mathbf{P}_{k|k-1} \mathbf{H}^T \in \mathbb{R}^{m \times m}$ is a symmetric matrix and $\Delta \mathbf{z}_{k|k-1} \in \mathbb{R}^{m \times 1}$, the number of inputs is $\frac{1}{2}m^2 + \frac{3}{2}m$. Besides, to ensure a positive estimate of $\frac{1}{\mathbb{E}(\lambda_k^{-1})}$ and reduce the range of FCN output values, $\exp(\cdot)$ is connected to the FCN output layer.

3) In Block III, we introduce a neural network parameter $\mathbf{R}_s \in \mathbb{R}^{n \times n}$, and the scale covariance matrix \mathbf{R} can be estimated by

$$(5.6) \quad \hat{\mathbf{R}} = (\varsigma_1 \mathbf{R}_s)(\varsigma_1 \mathbf{R}_s)^T,$$

where $\varsigma_1 > 1$ is used to increase the gradient of the loss function to \mathbf{R}_s . This allows for efficient updates of \mathbf{R}_s , particularly when the learning rate is small. Also, equation (5.6) guarantees the symmetry and positive semi-definiteness of $\hat{\mathbf{R}}$.

5.3 Unsupervised training algorithm

Following the introduction of the RKFnet, an unsupervised training method is proposed in this section, which can benefit various practical applications in scenarios where the ground-truth data is difficult to obtain. In [78], the unsupervised loss function is $\|\Delta \mathbf{z}_{k+1|k}\|^2$. Although achieving satisfactory performance in Gaussian noise scenarios, the employed L2 norm is sensitive to the outliers of the observation prediction error $\Delta \mathbf{z}_{k+1|k}$. Instead, we adopt the function $-\log(st(\cdot; v, \sigma))$, where $st(\cdot; v, \sigma)$ is a one-dimensional ST PDF with the dof value v and scale parameter σ . Given N_{traj} sequences with length T , the loss function can be written as

$$\mathbf{L}(\widetilde{\boldsymbol{\Theta}}) = \frac{1}{N_{traj} \times T \times m} \sum_{j=1}^{N_{traj}} \sum_{k=1}^T \sum_{i=1}^m st\left(\Delta \mathbf{z}_{k+1|k}^{i,j}(\widetilde{\boldsymbol{\Theta}}); v, \sigma\right) + \gamma_1 \|\hat{\mathbf{R}} - \mathbf{I}\|^2 + \gamma_2 \|\widetilde{\boldsymbol{\Theta}}\|^2,$$

where $\Delta \mathbf{z}_{k+1|k}^{i,j}$ is the i -th element of the observation prediction error at time k in the j -th trajectory, and $\widetilde{\boldsymbol{\Theta}}$ represents the parameters of the fully connected neural network in Block II. Also, due to the difficulty of manually giving the parameters of the ST PDF, we set $v = \exp(\varsigma_2 v')$ and $\sigma = \exp(\varsigma_3 \sigma')$, where v' and σ' are specified as neural network parameters, and $\varsigma_2 > 0$, $\varsigma_3 > 0$ are employed to increase the loss gradient related to v'

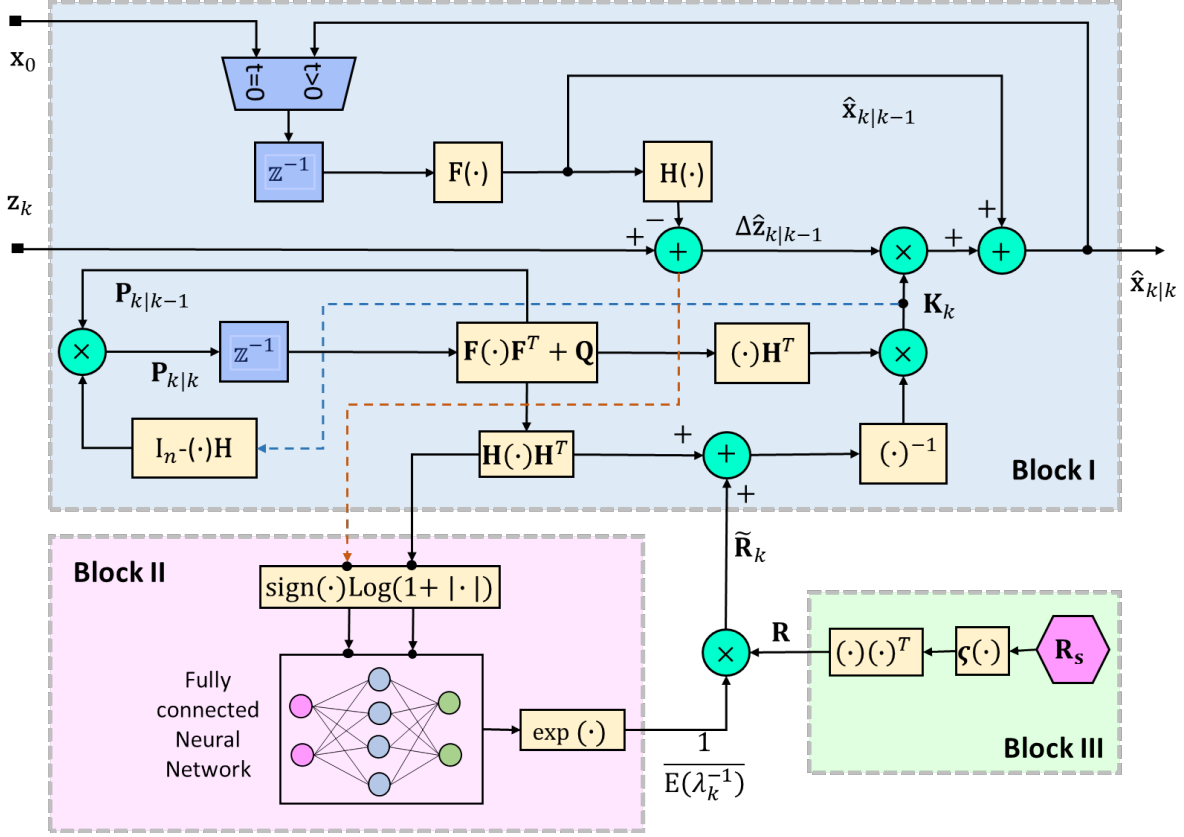


Figure 5.1: Structure of the RKFnet

and σ' . Besides, if $\hat{\mathbf{R}}$ is a solution, its scaled matrix $\psi\hat{\mathbf{R}}$, where $\psi > 0$, is also a solution, implying an infinite set of solutions. To keep the consistency of the training results, $\|\hat{\mathbf{R}} - 1\|^2$ with parameter γ_1 limits the value of $|\hat{\mathbf{R}}|$ close to 1. Furthermore, $\|\hat{\boldsymbol{\Theta}}\|^2$ is a penalty term with parameter γ_2 , and $\hat{\boldsymbol{\Theta}} = \{\hat{\boldsymbol{\Theta}}, v', \sigma', \mathbf{R}_s\}$.

The proposed RKFnet can be seen as a special RNN structure, where the temporally dependent information is delivered through Block I. Due to error accumulation, the training process of the RKFnet is not stable especially when the noise is highly heavy-tailed. Although the scheduled sampling technique [8] can improve the convergence stability of the sequence-to-sequence models, it cannot be directly employed in our framework due to its requirement for ground-truth data. Instead, we propose a USS method, of which the structure is shown in Figure 5.2. At any time k , the input of the RKFnet is chosen from either $[\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}]$ or the filtering result of a selected traditional RKF, $[\hat{\mathbf{x}}_{k-1|k-1}^*, \mathbf{P}_{k-1|k-1}^*]$ based on a coin toss. Also, to reduce the exposure bias caused by the differences between the training and inference processes, we employ the scheduled sampling method [8] and the probability of the RKF estimates, p_t , decreases linearly,

i.e.,

$$p_t = \max(p_{\min}, p_{\max} - \Delta p * t)$$

where t is the training iteration number and Δp represents the decreasing speed. Also, p_{\max} and p_{\min} are the maximal and minimal probability values of p_t , respectively.

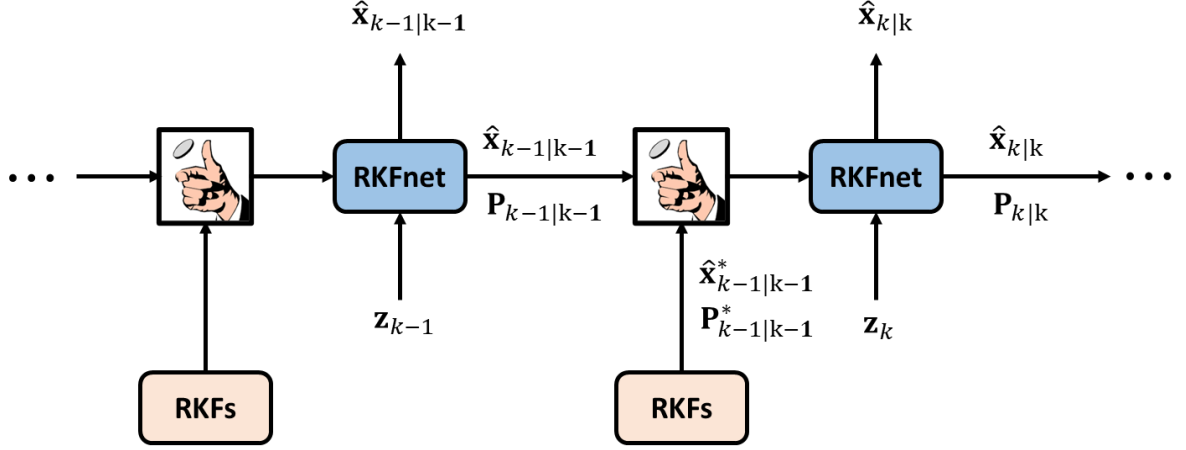


Figure 5.2: The unsupervised scheduled sampling technique.

5.4 Simulation results

5.4.1 Target-tracking models and traditional RKF's

The target-tracking model in the following experiments follows the same set as in Section 4.4, where a linear target-tracking model with various heavy-tailed noises is considered. Also, all the employed traditional RKF's contain the same parameters as in Section 4.4.5.

5.4.2 RKFnet parameters

In Block II, a 3-layer FCN is used in our tests, and the corresponding hidden-layer sizes are 32, 64 and 32, respectively. Also, the LeakyReLU function, characterized by a negative slope parameter of 0.1, is applied across all hidden layers. Furthermore, the initial values of each element in \mathbf{R}_s are drawn from a uniform distribution $U[0, \frac{1}{\varsigma_1}]$, and $\varsigma_1 = 300$. Besides, for the loss function, $\varsigma_2 = \varsigma_3 = 300$, $\gamma_1 = 0.1$ and $\gamma_2 = 0.0001$.

5.4.3 Training dataset and optimiser

In every tracking scenario, the training, cross-validation, and testing sets consist of 3200, 200, and 200 trajectories with length $T = 100$, respectively. Also, there are 2000 iterations in every training process, and we adopt the Adam optimiser, where the learning rate starts from 0.0002 and is halved every 400 iterations. The training batch size is 200 and for the USS, $p_{max} = 1$, $\Delta p = \frac{1}{600}$.

5.4.4 Performance evaluation in relation to p_{min} of USS

Based on the target-tracking model, we evaluate the influence of p_{min} , which takes its value from [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]. Also, the target tracking model with the SGaS measurement noise is considered, and the shape parameter is selected from [0.3, 0.7, 1.1, 1.5, 1.85]. The filtering results of the RKF-SGaS-GSIS are employed for the USS training method. To analyse the stability and consistency of the training results, 5 Monte Carlo experiments are run. As shown in Figure 5.3, due to the exposure bias, large p_{min} can cause worse filtering results, especially in the highly heavy-tailed noise scenario ($\alpha = 0.3$). By contrast, for small p_{min} , the estimation results are more precise and stable.

5.4.5 Comparison between the employed ST loss, L1 and L2

To ascertain the selected ST-based loss function, this experiment compares it with the classical L1 and L2 loss functions. The GM, ST and SGaS-based measurement noises are selected, and their shape parameters are selected from the same vectors as in Section 4.4.5. We set $p_{min} = 0$ and run 5 Monte Carlo experiments, where the optimal results are selected as the final results. As depicted in Figure 5.4, in the light-tailed noise scenarios, the filtering results based on the different loss functions are similar. In comparison, when the noise contains heavy outliers, the ST-based loss function obtains the best estimation due to its less sensitivity to the outliers.

5.4.6 RKFnet vs traditional RKFs

In this experiment, our proposed RKFnet is compared with three heavy-tailed-distribution-based RKFs, RSTKF, RKF-SL and RKF-SGaS. Also, the target tracking models are the same as in Section 4.4. Besides, to evaluate the influence of the filtering results from different RKFs on the RKFnet performance, all these three traditional RKFs are

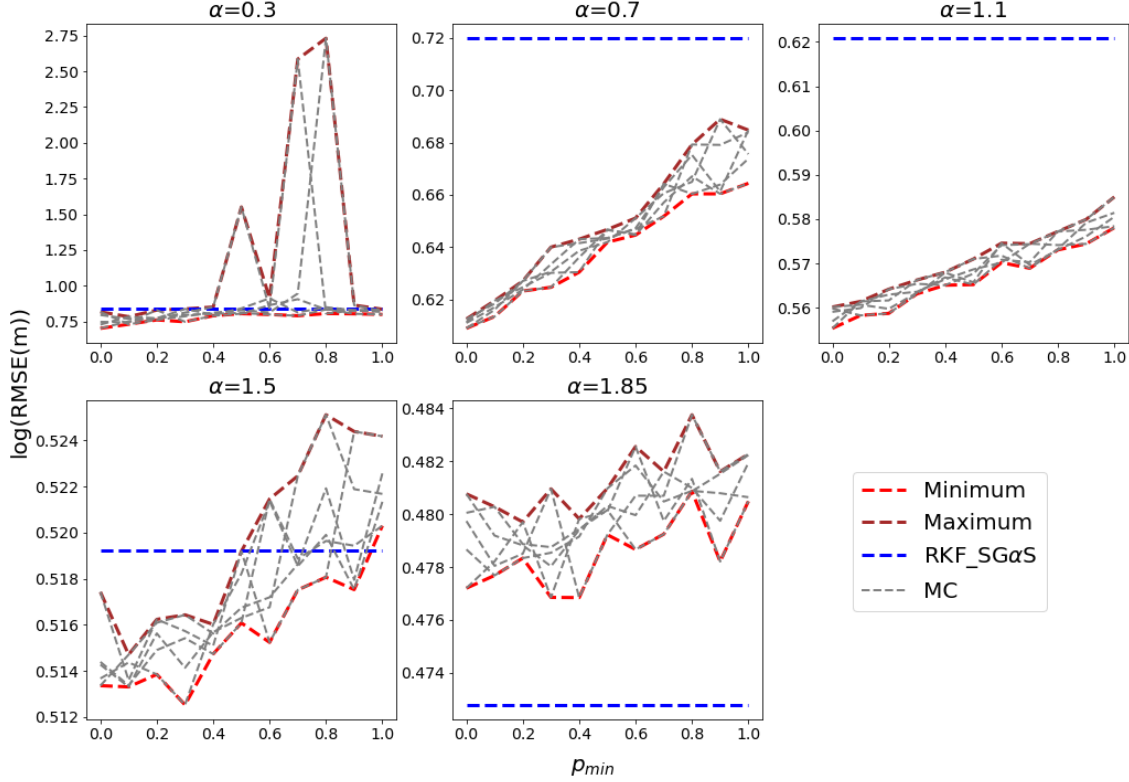


Figure 5.3: The tracking results of the RKFnet under the RKF-SG α S measurement noise. In every subplot, the title is the shape parameter α of the SG α S distribution, the x-axis represents p_{min} , and the y-axis is the position estimation error. Also, the "MC" represents the 5 independent Monte Carlo experiment results, and the corresponding maximal and minimal errors are depicted by "Maximum" and "Minimum", respectively. The filtering performance of the RKF-SG α S-GSIS is shown as a reference.

employed to produce the reference sequences. Besides, $p_{min} = 0$, and we run 5 Monte Carlo experiments, where the optimal results are selected as the final results. From Figure 5.5, when the noise distribution is light-tailed, the RKFnets achieves similar results to the traditional RKF. Nevertheless, the RKFnets perform better than the other traditional filters under the heavy-tailed noise scenarios, where larger model errors and the approximation of the VB method in the RKF frameworks degrade the filtering accuracy. Moreover, we should notice that the RKFnets based on the different traditional RKF obtain similar estimation results. This suggests that the RKFnet has no requirement for precise reference sequences, which is only employed to stabilise the convergence in the early stage of the training process. When the p_{min} gradually reduces towards zero, the RKFnet can converge to a satisfying position without the assistance of the RKF estimates.

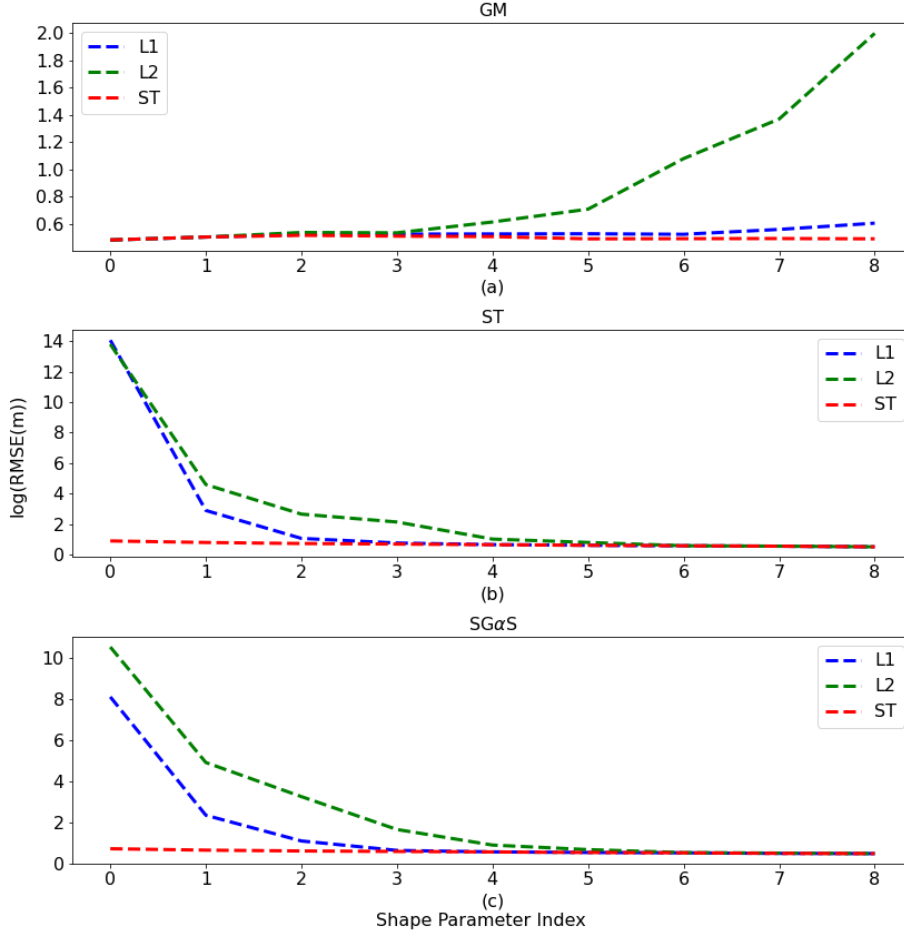


Figure 5.4: The filtering results of the RKFnet with different loss functions. The titles of the subplots are the distributions of the heavy-tailed measurement noise. The x-axis is the shape parameter index, and the y-axis represents the position estimation error. (a) plots the estimation errors under the GM noise, while the results under the ST and SGαS noise are shown in (b) and (c), respectively.

5.5 Conclusion

In this chapter, we presented a DD-MB RKF framework, RKFnet, combining the traditional heavy-tailed-distribution-based RKF and a deep-learning technique. Specifically, the mixing-parameter-based function is estimated by an FCN, and the scale matrix can be calculated based on an introduced neural network parameter. Also, the USS training method is proposed to improve the stability of the training process. Besides, a comparison between the proposed RKFnet and three heavy-tailed-distribution-based filters was conducted. The experimental results show that in the light-tailed scenarios, all the filters

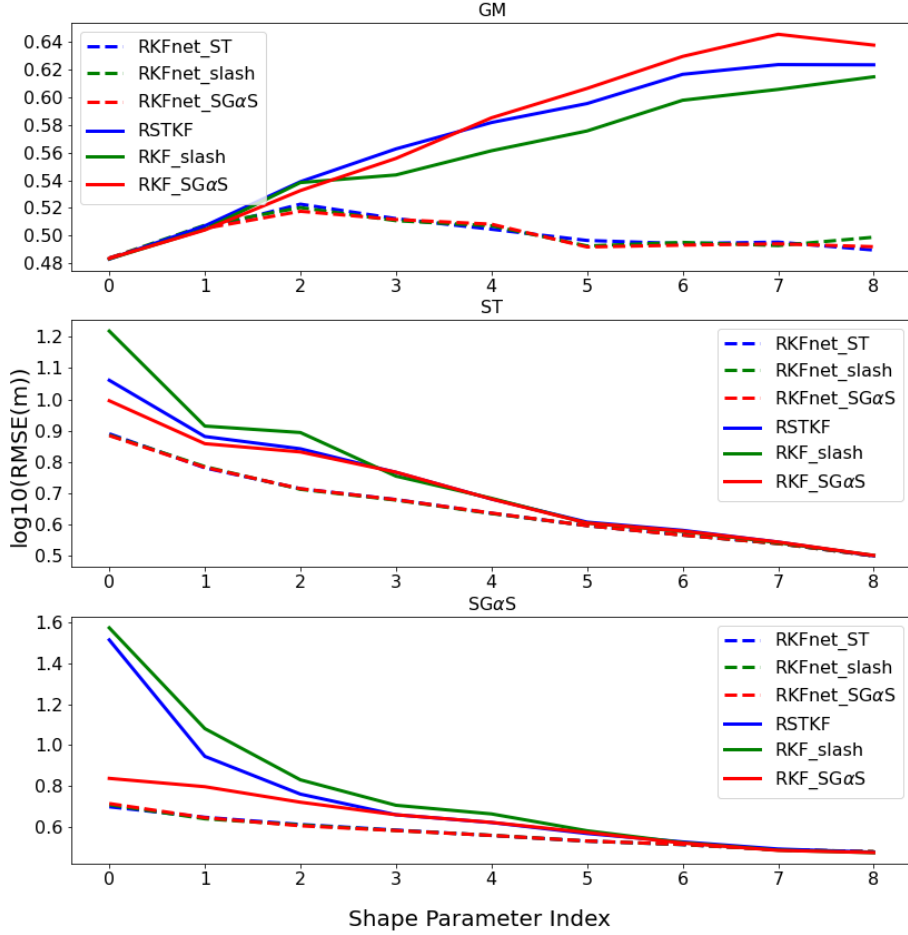


Figure 5.5: The filtering performance comparison between the RKFnet and traditional RSTKF, RKF-slash and RKF-SGαS. The titles of the subplots are the distributions of the heavy-tailed measurement noise. The x-axis is the shape parameter index, and the y-axis represents the position estimation error. (a) plots the estimation errors under the GM noise, while the results under the ST and SGαS noise are shown in (b) and (c), respectively.

produce similar results. However, the RKFnet achieves the best performance under various heavy-tailed noises. In particular, the performance of the RKFnet does not hinge on accurate filtering results from the traditional filters. Although these outcomes are employed in the early USS training stage, the RKFnet can be trained in a free-running manner when p_t is gradually reduced to zero.

In the future, we will consider applying the proposed framework to more complicated scenarios. In this work, we focus on linear models with heavy-tailed symmetric measurement noise. However, an extension to the linear models with skewed heavy-tailed

signal and measurement noises can also be considered, which can benefit more practical applications.

CONCLUSION

This thesis is dedicated to enhancing sequential Bayesian filters, offering advancements applicable across a multitude of scenarios.

Considering particle degeneracy of the standard PF, Chapter 3 provides our proposed novel hybrid filtering approach, the PSMF, using ideas from SMF. Also, two variants of the PSMF, PSMF-L and PSMF-NL, are presented, adopting linear and nonlinear transport maps, respectively. Besides, two sets of experiments involving four widely employed state-space models were implemented to validate the proposed hybrid filters. The first experimental setup shows that for medium and large numbers of particles, the PSMF-L yields better results than the benchmark approaches. Nevertheless, in some cases, the PSMF-NL can provide more accurate estimation than to the PSMF-L, where the linear approximation of the linear map causes large errors. By contrast, the second experiment demonstrates that the PSMF-NL is more tolerant to parameter changes than the PSMF-L, benefiting from the flexibility of nonlinear maps.

Although the PSMF presents robust filtering performance, it suffers from expensive computational costs due to its Monte Carlo sampling strategy. Instead, Chapter 4 focuses on the RKF framework, providing efficient solution under linear models with heavy-tailed noise. Specifically, we proposed a new robust Kalman filter framework based on the SG α S distribution, RKF-SG α S, and then its four variants are presented based on the four MMSE estimators of the scale function. Also, the comparison among these variants is implemented in the first experiment, and the RKF-SG α S-GSIS obtains the most efficient performance. In the second simulation, the RKF-SG α S is compared with the

state-of-art RKF, and its filtering advantage under the SG α S noise is shown. Besides, although our RKF-SG α S is more computationally expensive than the benchmark filters, their computational burdens become closer when the noise distribution is heavy-tailed.

To alleviate the reliance of the proposed RKF-SG α S on precise state-space models, we present a new neural network architecture, combining the traditional RKF methods with the deep learning technique and hence called RKFnet, which is based on the new hierarchical Gaussian state-space model. Also, the USS training method is proposed to improve the stability of the training process. To validate our proposed work, three experiments are conducted. The first two experiments clarify the selections on the USS parameters and the loss function. By contrast, the last experiment demonstrates that the RKFnet can performance better than the traditional filters under the heavy-tailed noise.

6.1 Discussion and Future Work

In this thesis, we have presented novel filtering approaches to address challenges in target tracking scenarios. Our proposed filters have shown promising results and opened up various avenues for future research and improvements. The following points outline the potential directions for future work:

1. The PSMF extension to higher-Dimensional models: The PSMF has exhibited impressive performance in low-dimensional models. To enhance its practicality, further research can be directed toward higher-dimensional problems [75, 96]. Additionally, investigating the integration of the DMF [87] with our hybrid filters presents an exciting avenue for exploration.
2. The applications of the RKF-SG α S and RKFnet to state-space models for complex scenarios: The proposed RKF frameworks have primarily dealt with linear models characterized by heavy-tailed symmetric measurement noise. To make our framework even more versatile, future work will aim to extend the RKF filters to linear/nonlinear models with skewed heavy-tailed signal and measurement noises. This will enable our filters to cater to a broader range of practical applications.

BIBLIOGRAPHY

- [1] *Expectations anchoring indexes for brazil using kalman filter: Exploring signals of inflation anchoring in the long term*, International Economics, 165 (2020), pp. 72–91.
- [2] C. A, T. J, D. G, AND D. A., *Differentiable particle filtering via entropy-regularized optimal transport*, In Proceedings of the 38th International Conference on Machine Learning, 139 (2021), p. 2100–2111.
- [3] A. ACHIM, B. BUXTON, G. TZAGKARAKIS, AND P. TSAKALIDES, *Compressive sensing for ultrasound rf echoes using α -stable distributions*, 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology, (2010).
- [4] G. AGAMENNONI, J. I. NIETO, AND E. M. NEBOT, *Approximate inference in state-space models with heavy-tailed noise*, IEEE Transactions on Signal Processing, 60 (2012), p. 5024–5037.
- [5] P. AGGARWAL, D. GU, S. NASSAR, Z. SYED, AND N. EL-SHEIMY, *Extended particle filter (epf) for ins/gps land vehicle navigation applications*, In Proceedings of the Institute of Navigation Satellite Division Technical Meeting (ION GNSS 2007), (2007), p. 2619–2626.
- [6] M. S. ARULAMPALAM, S. MASKELL, N. GORDON, AND T. CLAPP, *A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking*, IEEE Transactions on signal processing, 50 (2002), pp. 174–188.
- [7] M. BAI, Y. HUANG, B. CHEN, L. YANG, AND Y. ZHANG, *A novel mixture distributions-based robust kalman filter for cooperative localization*, IEEE Sensors Journal, 20 (2020), p. 14994–15006.

- [8] S. BENGIO, O. VINYALS, N. JAITLEY, AND N. SHAZEER, *Scheduled sampling for sequence prediction with recurrent neural networks*, Advances in neural information processing systems, 28 (2015).
- [9] T. BENGTTSSON, P. BICKEL, AND B. LI, *Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems*, Institute of Mathematical Statistics Collections, 2 (2008), pp. 316–334.
- [10] H. BERGSTRÖM, *On some expansions of stable distribution functions*, Arkiv för Matematik, 2 (1952), pp. 375–378.
- [11] G. BURGERS, P. J. VAN LEEUWEN, AND G. EVENSEN, *Analysis scheme in the ensemble kalman filter*, Monthly Weather Review, 126 (1998), p. 1719–1724.
- [12] K. BURNECKI, A. WYŁOMAŃSKA, A. BELETSKII, V. GONCHAR, AND A. CHECHKIN, *Recognition of stable distribution with lévy index α close to 2*, Physical Review E, 85 (2012), p. 056711.
- [13] O. CAPPÉ, S. J. GODSILL, AND E. MOULINES, *An overview of existing methods and recent advances in sequential monte carlo*, Proceedings of the IEEE, 95 (2007), pp. 899–924.
- [14] C. CARMELI, E. DE VITO, AND A. TOIGO, *Vector valued reproducing kernel hilbert spaces of integrable functions and mercer theorem*, Analysis and Applications, 04 (2006), p. 377–408.
- [15] F. CASSOLA AND M. BURLANDO, *Wind speed and wind energy forecast through kalman filtering of numerical weather prediction model output*, Applied Energy, 99 (2012), p. 154–166.
- [16] J. CHEN, H. JING, Y. CHANG, AND Q. LIU, *Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process*, Reliability Engineering and System Safety, 185 (2019), p. 372–382.
- [17] Z. CHEN, *Bayesian filtering: From kalman filters to particle filters, and beyond*, Statistics, 182 (2003), pp. 1–69.
- [18] N. CHUSTAGULPROM, S. REICH, AND M. REINHARDT, *A hybrid ensemble transform particle filter for nonlinear and spatially extended dynamical systems*, SIAM/ASA J. UNCERTAINTY QUANTIFICATION, 4 (2016), pp. 592–608.

- [19] H. COSKUN, F. ACHILLES, R. DIPIETRO, N. NAVAB, AND F. TOMBARI, *Long short-term memory kalman filters: Recurrent neural estimators for pose regularization*, 2017 IEEE International Conference on Computer Vision (ICCV), (2017).
- [20] C. COTTER, S. COTTER, AND P. RUSSELL, *Ensemble transport adaptive importance sampling*, SIAM/ASA Journal on Uncertainty Quantification, 7 (2019), pp. 444–471.
- [21] L. DELLE MONACHE, T. NIPEN, Y. LIU, G. ROUX, AND R. STULL, *Kalman filter and analog schemes to postprocess numerical weather predictions*, Monthly Weather Review, 139 (2011), p. 3554–3570.
- [22] P. DONG, Z. JING, H. LEUNG, K. SHEN, AND J. WANG, *Student-t mixture labeled multi-bernoulli filter for multi-target tracking with heavy-tailed noise*, Signal Processing, (2018), pp. 331–339.
- [23] A. DOUCET, S. GODSILL, AND C. ANDRIEU, *On sequential monte carlo sampling methods for bayesian filtering*, Statistics and computing, 10 (2000), pp. 197–208.
- [24] G. EVENSEN, *Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistic*, JOURNAL OF GEOPHYSICAL RESEARCH, 99 (1994), pp. 10,143–10,162.
- [25] C. W. FOX AND S. J. ROBERTS, *A tutorial on variational bayesian inference*, Artificial Intelligence Review, 38 (2011), p. 85–95.
- [26] M. FREI AND H. R. KUNSCH, *Bridging the ensemble kalman and particle filters*, Biometrika, 100 (2013), pp. 781–800.
- [27] R. GAN, B. I. AHMAD, AND S. J. GODSILL, *Lévy state-space models for tracking and intent prediction of highly maneuverable objects*, IEEE Transactions on Aerospace and Electronic Systems, 57 (2021).
- [28] C. GAO, J. YAN, S. ZHOU, B. CHEN, AND H. LIU, *Long short-term memory-based recurrent neural networks for nonlinear target tracking*, Signal Processing, 164 (2019), p. 67–73.
- [29] S. GODSILL AND E. E. KURUOGLU, *Bayesian inference for time series with heavy-tailed symmetric α -stable noise processes*, Proc. Applications of heavy tailed distributions in economics, engineering and statistics, (1999).

- [30] ———, *Bayesian inference for time series with heavy-tailed symmetric α -stable noise processes*, Proc. Applications of heavy tailed distributions in economics, engineering and statistics, (1999).
- [31] I. GROOMS AND G. ROBINSON, *A hybrid particle-ensemble kalman filter for problems with medium nonlinearity*, PLOS ONE, 16 (2021).
- [32] W. GUO, C. HAN, AND M. LEI, *Improved unscented particle filter for nonlinear bayesian estimation*, 2007 10th International Conference on Information Fusion, (2007).
- [33] P. HAO, O. KARAKUŞ, AND A. ACHIM, *A hybrid particle-stochastic map filter*, Signal Processing, 207 (2023), p. 108969.
- [34] P. HAO, O. KARAKUŞ, AND A. ACHIM, *Robust kalman filters based on the sub-gaussian α -stable distribution*, 2023.
- [35] D. HENKE, E. M. DOMINGUEZ, D. SMALL, M. E. SCHAEPMAN, AND E. MEIER, *Moving target tracking in single- and multichannel sar*, IEEE Transactions on Geoscience and Remote Sensing, 53 (2015), pp. 3146–3159.
- [36] D. HENKE, C. MAGNARD, M. FRILOUD, D. SMALL, E. MEIER, AND M. SCHAEPMAN, *Moving-target tracking in single-channel wide-beam sar*, IEEE Transactions on Geoscience and Remote Sensing, 50 (2012), pp. 4735–4747.
- [37] J. D. HOL, T. B. SCHON, AND F. GUSTAFSSON, *On resampling algorithms for particle filters*, 2006 IEEE Nonlinear Statistical Signal Processing Workshop, (2006).
- [38] Y. HUANG, Y. ZHANG, N. LI, AND J. CHAMBERS, *Robust student's t based nonlinear filter and smoother*, IEEE Transactions on Aerospace and Electronic Systems, 52 (2016), p. 2586–2596.
- [39] Y. HUANG, Y. ZHANG, N. LI, Z. WU, AND J. A. CHAMBERS, *A novel robust student's t -based kalman filter*, IEEE Transactions on Aerospace and Electronic Systems, 53 (2017), p. 1545–1554.
- [40] Y. HUANG, Y. ZHANG, P. SHI, Z. WU, J. QIAN, AND J. A. CHAMBERS, *Robust kalman filters based on gaussian scale mixture distributions with application to target tracking*, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, 49 (2019), pp. 2082–2096.

-
- [41] Y. HUANG, Y. ZHANG, P. SHI, Z. WU, J. QIAN, AND J. A. CHAMBERS, *Robust kalman filters based on gaussian scale mixture distributions with application to target tracking*, IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49 (2019), p. 2082–2096.
- [42] Y. HUANG, Y. ZHANG, Y. ZHAO, AND J. A. CHAMBERS, *A novel robust gaussian–student’s t mixture distribution based kalman filter*, IEEE Transactions on Signal Processing, 67 (2019), p. 3606–3620.
- [43] Y. HUANG, Y. ZHANG, Y. ZHAO, L. MIHAYLOVA, AND J. CHAMBERS, *A novel robust rauch-tung-striebeel smoother based on slash and generalized hyperbolic skew student’s t -distributions*, in 2018 21st International Conference on Information Fusion (FUSION), IEEE, 2018, pp. 369–376.
- [44] Y. HUANG, Y. ZHANG, Y. ZHAO, P. SHI, AND J. A. CHAMBERS, *A novel outlier-robust kalman filtering framework based on statistical similarity measure*, IEEE Transactions on Automatic Control, 66 (2021), p. 2677–2692.
- [45] Y. HUANG, F. ZHU, Y. ZHANG, Y. ZHAO, P. SHI, AND J. CHAMBERS, *4 outlier-robust kalman filtering framework based on statistical similarity measure*, Outliers in Control Engineering, (2022), p. 61–98.
- [46] R. IZANLOO, S. A. FAKOORIAN, H. S. YAZDI, AND D. SIMON, *Kalman filtering based on the maximum correntropy criterion in the presence of non-gaussian noise*, 2016 Annual Conference on Information Science and Systems (CISS), (2016).
- [47] S. JULIER, *The scaled unscented transformation*, Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301), (2002).
- [48] S. JULIER AND J. K. UHLMANN, *A general method for approximating nonlinear transformations of probability distributions*, Citeseer, (1996).
- [49] S. J. JULIER AND J. K. UHLMANN, *A new extension of the kalman filter to nonlinear systems*, Signal processing, sensor fusion, and target recognition VI, 3068 (1997), pp. 182–193.
- [50] R. E. KALMAN, *A new approach to linear filtering and prediction problems*, Transactions of the ASME—Journal of Basic Engineering, 82 (1960), pp. 35–45.

- [51] C. D. KARLGAARD AND H. SCHAUB, *Huber-based divided difference filtering*, Journal of Guidance, Control, and Dynamics, 30 (2007), p. 885–891.
- [52] M. G. KENDALL, A. STUART, AND J. K. ORD, *Kendall's advanced theory of Statistics*, Hodder Arnold, 1994.
- [53] S. KOLOURI, S. R. PARK, M. THORPE, D. SLEPCEV, AND G. K. ROHDE, *Optimal mass transport: Signal processing and machine-learning applications*, IEEE signal processing magazine, 34 (2017), pp. 43–59.
- [54] N. KOVVALI, *Theory and applications of gaussian quadrature methods*, Springer Cham, 2011.
- [55] S. KRING, S. T. RACHEV, M. HÖCHSTÖTTER, AND F. J. FABOZZI, *Estimation of α -stable sub-gaussian distributions for asset returns*, Contributions to Economics, p. 111–152.
- [56] G. Y. KULIKOV AND M. KULIKOVA, *Estimation of maneuvering target in the presence of non-gaussian noise: A coordinated turn case study*, Signal Processing, (2018), pp. 241–257.
- [57] Y. LECUN, Y. BENGIO, AND G. HINTON, *Deep learning*, Nature, 521 (2015), p. 436–444.
- [58] T. LEMKE, M. RIABIZ, AND S. J. GODSILL, *Fully bayesian inference for α -stable distributions using a poisson series representation*, Digital Signal Processing, 47 (2015), pp. 96–115.
- [59] T. LI, M. BOLIC, AND P. M. DJURIC, *Resampling methods for particle filtering: classification, implementation, and strategies*, IEEE Signal processing magazine, 32 (2015), pp. 70–86.
- [60] Y. LIU, J. CHENG, H. ZHANG, H. ZOU, AND N. XIONG, *Long short-term memory networks based on particle filter for object tracking*, IEEE Access, 8 (2020), p. 216245–216258.
- [61] Y. MARZOUK, T. MOSELHY, M. PARNO, AND A. SPANTINI, *Sampling via measure transport: An introduction*, (eds) Handbook of Uncertainty Quantification, (2016).

- [62] L. MIHAYLOVA, P. BRASNETT, A. ACHIM, D. BULL, AND N. CANAGARAJAH, *Particle filtering with alpha-stable distributions*, IEEE/SP 13th Workshop on Statistical Signal Processing, (2015).
- [63] L. MIHAYLOVA, A. Y. CARMÍ, F. SEPTIER, A. GNING, S. K. PANG, AND S. GODSILL, *Overview of bayesian sequential monte carlo methods for group and extended object tracking*, Digital Signal Processing, 25 (2014), pp. 1–16.
- [64] S. M. MOHIUDDIN AND J. QI, *Maximum correntropy extended kalman filtering for power system dynamic state estimation*, 2019 IEEE Power & Energy Society General Meeting (PESGM), (2019).
- [65] T. A. E. MOSELHY AND Y. M. MARZOUK, *Bayesian inference with optimal maps*, Journal of Computational Physics, 231 (2012), pp. 7815–7850.
- [66] R. MUNGUÍA, J. DAVALOS, AND S. URZUA, *Estimation of the solow-cobb-douglas economic growth model with a kalman filter: An observability-based approach*, Heliyon, 5 (2019).
- [67] J. NERI, P. DEPALLE, AND R. BADEAU, *Approximate inference and learning of state space models with laplace noise*, IEEE Transactions on Signal Processing, 69 (2021), p. 3176–3189.
- [68] J. P. NOLAN, *Numerical calculation of stable densities and distribution functions*, Communications in Statistics. Stochastic Models, 13 (1997), p. 759–774.
- [69] ———, *Maximum likelihood estimation and diagnostics for stable distributions*, Lévy Processes, (2001), p. 379–400.
- [70] J. P. NOLAN, *Models for heavy tailed data*, in Univariate stable distributions, Springer Cham, 2020, pp. 1–333.
- [71] J. P. NOLAN, J. G. GONZALEZ, AND R. C. NÚÑEZ, *Stable filters: A robust signal processing framework for heavy-tailed noise*, in 2010 IEEE Radar Conference, IEEE, 2010, pp. 470–473.
- [72] H. NURMINEN, T. ARDESHIRI, R. PICHE, AND F. GUSTAFSSON, *Skew-t filter and smoother with improved covariance matrix approximation*, IEEE Transactions on Signal Processing, 66 (2018), p. 5618–5633.

- [73] N. PAPADAKIS, E. MÉMIN, A. CUZOL, AND N. GENGEMBRE, *Data assimilation with the weighted ensemble kalman filter*, Tellus A, 62 (2010), pp. 673–697.
- [74] M. D. PARNO AND Y. M. MARZOUK, *Transport map accelerated Markov chain Monte Carlo*, SIAM/ASA Journal on Uncertainty Quantification, 6 (2018), pp. 645–682.
- [75] J. POTERJOY, *A localized particle filter for high-dimensional nonlinear systems*, Monthly Weather Review, 144 (2016), pp. 59–76.
- [76] M. PULIDOAB AND P. J. VAN LEEUWEN, *Sequential monte carlo with kernel embedded mappings: The mapping particle filter*, Journal of Computational Physics, 396 (2019), pp. 400–415.
- [77] S. REICH, *A nonparametric ensemble transform method for bayesian inference*, SIAM J. SCI. COMPUT., 35 (2013), pp. A2013–A2024.
- [78] G. REVACH, N. SHLEZINGER, T. LOCHER, X. NI, R. J. VAN SLOUN, AND Y. C. ELDAR, *Unsupervised learned kalman filtering*, 2022 30th European Signal Processing Conference (EUSIPCO), (2022).
- [79] G. REVACH, N. SHLEZINGER, X. NI, A. L. ESCORIZA, R. J. VAN SLOUN, AND Y. C. ELDAR, *Kalmanet: Neural network aided kalman filtering for partially known dynamics*, IEEE Transactions on Signal Processing, 70 (2022), p. 1532–1547.
- [80] S. ROBERT AND H. R. KÜNSCH, *Localizing the ensemble kalman particle filter*, Tellus A: Dynamic Meteorology and Oceanography, 69 (2017).
- [81] M. ROTH, E. OZKAN, AND F. GUSTAFSSON, *A student’s t filter for heavy tailed process and measurement noise*, 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, (2013).
- [82] P. SAKOV AND P. R. OKE, *Implications of the form of the ensemble transformation in the ensemble square root filters*, Monthly Weather Review, 136 (2008), p. 1042–1053.
- [83] H. SALEHINEJAD, S. SANKAR, J. BARFETT, E. COLAK, AND S. VALAEE, *Recent advances in recurrent neural networks*, Feb 2018.
- [84] G. SAMORODNITSKY AND M. S. TAQQU, *Stable non-gaussian random processes*, (2017).

-
- [85] G.-H. SEO, B.-J. CHOI, Y.-K. CHO, Y. H. KIM, AND S. KIM, *Assimilation of sea surface temperature in the northwest pacific ocean and its marginal seas using the ensemble kalman filter*, Ocean Science Journal, 45 (2010), p. 225–242.
- [86] S. SINGH, W. RICHARDS, J. ZINECKER, AND D. WILTON, *Accelerating the convergence of series representing the free space periodic green’s function*, IEEE Transactions on Antennas and Propagation, 38 (1990), p. 1958–1962.
- [87] A. SPANTINI, R. BAPTISTA, AND Y. MARZOUK, *Coupling techniques for nonlinear ensemble filtering*, SIAM Review, 64 (2022), p. 921–953.
- [88] J. STEWART, D. K. CLEGG, AND S. WATSON, *Calculus: early transcendentals*, Cengage Learning, 2020.
- [89] A. S. STORDAL, H. A. KARLSEN, G. NÆVDAL, H. J. SKAUG, AND B. VALLÈS, *Bridging the ensemble kalman filter and particle filters: the adaptive gaussian mixture filter*, Comput Geosci, 15 (2011), pp. 293–305.
- [90] S. P. TALEBI, S. J. GODSILL, AND D. P. MANDIC, *Filtering structures for α -stable systems*, IEEE Control Systems Letters, 7 (2022), pp. 553–558.
- [91] M. TEIMOURI, S. REZAKHAH, AND A. MOHAMMADPOUR, *Parameter estimation using the em algorithm for symmetric stable random variables and sub-gaussian random vectors*, Journal of Statistical Theory and Applications, 17 (2018), p. 439.
- [92] M. K. TIPPETT, J. L. ANDERSON, C. H. BISHOP, T. M. HAMILL, AND J. S. WHITAKER, *Ensemble square root filters*, Monthly Weather Review, 131 (2003), p. 1485–1490.
- [93] S. T. TOKDAR AND R. E. KASS, *Importance sampling: A review*, WIREs Computational Statistics, 2 (2009), p. 54–60.
- [94] C.-H. TSENG, S.-F. LIN, AND D.-J. JWO, *Robust huber-based cubature kalman filter for gps navigation processing*, Journal of Navigation, 70 (2016), p. 527–546.
- [95] D. G. TZIKAS, A. C. LIKAS, AND N. P. GALATSANOS, *The variational approximation for bayesian inference*, IEEE Signal Processing Magazine, 25 (2008), p. 131–146.

- [96] P. J. VAN LEEUWEN, H. R. KÜNSCH, L. NERGER, R. POTTHAST, AND S. REICH, *Particle filters for high-dimensional geoscience applications: A review*, Quarterly Journal of the Royal Meteorological Society, 145 (2019), pp. 2335–2365.
- [97] C. VILLANI, *Optimal transport, old and new*, Springer, Berlin Heidelberg New York, 2008.
- [98] W. WAN, J. FENG, B. SONG, AND X. LI, *Huber-based robust unscented kalman filter distributed drive electric vehicle state observation*, Energies, 14 (2021), p. 750.
- [99] G. WANG, C. YANG, AND X. MA, *A novel robust nonlinear kalman filter based on multivariate laplace distribution*, IEEE Transactions on Circuits and Systems II: Express Briefs, 68 (2021), p. 2705–2709.
- [100] G. WANG, Y. ZHANG, AND X. WANG, *Iterated maximum correntropy unscented kalman filters for non-gaussian systems*, Signal Processing, 163 (2019), p. 87–94.
- [101] G. WANG, Y. ZHANG, AND X. WANG, *Iterated maximum correntropy unscented kalman filters for non-gaussian systems*, Signal Processing, 163 (2019), pp. 87–94.
- [102] A. WERON AND R. WERON, *Computer simulation of lévy alpha-stable variables and processes*, Chaos — The Interplay Between Stochastic and Deterministic Behaviour, p. 379–392.
- [103] Y.-C. WU AND J.-W. FENG, *Development and application of artificial neural network*, Wireless Personal Communications, 102 (2017), p. 1645–1656.
- [104] C. XUE, Y. HUANG, F. ZHU, Y. ZHANG, AND J. A. CHAMBERS, *An outlier-robust kalman filter with adaptive selection of elliptically contoured distributions*, IEEE Transactions on Signal Processing, 70 (2022), p. 994–1009.
- [105] C. YARDIM, Z.-H. MICHALOPOULOU, AND P. GERSTOFT, *An overview of sequential bayesian filtering in ocean acoustics*, IEEE Journal of Oceanic Engineering, 36 (2011), p. 71–89.
- [106] Z. ZHANG, M. HOU, F. ZHANG, AND C. R. EDWARDS, *An lstm based kalman filter for spatio-temporal ocean currents assimilation*, Proceedings of the International Conference on Underwater Networks and Systems, (2019).

- [107] H. ZHU, H. LEUNG, AND Z. HE, *A variational bayesian approach to robust sensor fusion based on student-t distribution*, Information Sciences, 221 (2013), p. 201–214.
- [108] H. ZHU, G. ZHANG, Y. LI, AND H. LEUNG, *An adaptive kalman filter with inaccurate noise covariances in the presence of outliers*, IEEE Transactions on Automatic Control, 67 (2022), p. 374–381.

