Distributed Ensemble Kalman Filtering

Arslan Shahid

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

March 2014

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Engineering.

Abstract

Distributed estimation in a wireless sensor network has many advantages. It eliminates the need of the centralized knowledge of the measurement model parameters and does not have a single point of failure. Also, any sensor node can be queried to retrieve the state estimate. Furthermore, for high dimensional measurements, local processing of information also results in a significant reduction in the communication overhead.

The existing distributed Kalman filtering schemes have good computational and communication efficiency, but do not work well for non-linear and non-Gaussian problems. On the other hand, the distributed particle filtering schemes handle the non-linearities well, but have a much higher computational and communication cost.

In this thesis, we propose novel distributed filtering schemes based on the ensemble Kalman filter (EnKF). We consider three forms of the EnKF and express their update equations in an alternative information form. This allows us to use the randomized gossip algorithm to reach consensus on the sufficient statistics and perform local updates. The simulation results show that all three forms of the EnKF have a considerably lower computational cost compared to the particle filters. The results suggest that the proposed distributed schemes outperform the existing state-of-the-art distributed filtering schemes in two scenarios, a) linear measurement model with non-linear state dynamics, and b) high dimensional measurements (model parameters known everywhere in the network) with non-linear measurement model and state dynamics. In both of these scenarios, the proposed schemes achieve an estimation accuracy comparable to the existing state-of-the-art schemes while significantly reducing the communication cost.

Résumé

L'estimation distribuée dans un réseau de capteurs sans fil possède plusieurs avantages. Elle élimine le besoin d'une connaissance centralisée des paramtres du modèle de mesure et n'a pas un point de défaillance unique. Aussi, n'importe quel agent-capteur peut-être consulté pour obtenir une approximation de l'état général. De plus, pour des mesures de hautes dimensions, le calcul local d'informations résulte en une réduction significative des coûts de communication.

Les implémentations courantes du filtre de Kalman sont efficaces sur les plans de la charge de calcul et de la communication mais ne le sont pas pour les problmes non-linéaires et non-gaussiens. D'un autre côté, les techniques distribuées de filtrage particulaire gèrent avec succès les cas non-linéaires mais sont coûteuses sur les plans de la charge de calcul et de la communication.

Dans cette thèse, nous proposons des techniques de filtrage distribué basées sur le filtre de Kalman d'ensemble (FKEn). Nous considérons trois formes du FKEn et exprimons leurs équations de changement sous une forme alternative. Cela nous permet d'utiliser un algorithme de gossip aléatoire afin d'atteindre un consensus sur les statistiques suffisantes et calculer les changements locaux. Les résultats des simulations montrent que les trois formes de FKEn ont une charge de calcul bien moindre que les filtres de particules équivalents. Les rèsultats suggèrent que les techniques de filtrage distribué proposées sont plus efficaces que celles de pointe pour deux scénarios: a) un modèle de mesure linéaire avec des dynamiques d'états non-linéaires et b) des mesures de hautes dimensions (les paramtres du modèle sont connus de chaque agent) avec un modle de mesure non-linéaire et des dynamiques d'états non-linéaires. Dans les deux scénarios considérés, les techniques proposées atteignent une précision d'estimation comparable à celle des techniques de pointe tout en réduisant significativement les coûts de communication.

Acknowledgments

I'm greatly thankful to my supervisor, Prof. Mark Coates, for his guidance and continuous support. His vision, domain knowledge, expert advice and constructive feedback have immensely helped me in my research and during the writing of this thesis.

I'm also thankful to Prof. Michael Rabbat, for his help and guidance during my stay in the Computer Networks Group.

I would also like to thank my colleagues in the Computer Networks Group, Rizwan, Santosh, Alok, Yunpeng, Zhe, Milad, Guillaume, Deniz, Benjamin, Konstantinos, and Shohreh for helping and motivating me. I would specially thank Santosh and Deniz for their help and advice related to the algorithmic implementations. Also, a special thanks to Guillaume for translating my abstract into French.

Contents

List of Figures

List of Tables

List of Acronyms

WSN	Wireless Sensor Network
KF	Kalman Filter
EKF	Extended Kalman Filter
EnKF	Ensemble Kalman Filter
ESRF	Ensemble Square Root Filter
DEnKF	Deterministic Ensemble Kalman Filter
UKF	Unscented Kalman Filter
PF	Particle Filter
BPF	Bootstrap Particle Filter
APF	Auxiliary Particle Filter
SIR	Sequential Importance Resampling
GMM	Gaussian Mixture Model
MC	Monte Carlo
MCMC	Markov Chain Monte Carlo
RF	Radio Frequency
RSS	Received Signal Strength
RMS	Root Mean Square
RMSE	Root Mean Square Error

Chapter 1

Introduction

1.1 Problem Statement

This thesis considers the problem of distributed state estimation in wireless sensor networks (WSNs). In this problem, the goal is to sequentially estimate the system state that evolves according to a discrete-time Markov process. The evolution of the state is characterized by the *system model*

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}) + \mathbf{n}_{k-1}, \tag{1.1}$$

where $\mathbf{x} \in \mathcal{R}^n$ is the system state, $f : \mathcal{R}^n \to \mathcal{R}^n$ is a nonlinear system function and $\mathbf{n}_{k-1} \in \mathcal{R}^n$ is the system noise which follows a Gaussian distribution with zero mean and covariance $\mathbf{Q}$.

A set of wireless sensor nodes V are deployed in a physical region and record measurements related to the system state. The relationship between the local measurement of a sensor node $v \in V$ and the state is specified by its *measurement model*

$$\mathbf{y}^v = h^v(\mathbf{x}) + \epsilon^v, \quad v \in V, \tag{1.2}$$

where $\mathbf{y}^v \in \mathcal{R}^p$, $h^v : \mathcal{R}^n \to \mathcal{R}^p$ and $\epsilon_k^v \in \mathcal{R}^p$ are the sensor-dependent measurement, measurement function (possibly non-linear) and measurement noise, respectively. The measurement noise for each sensor node $v \in V$ follows a Gaussian distribution with mean zero and covariance $\mathbf{R}^v$ and the sensor nodes have uncorrelated measurement noise.

We assume that the sensor nodes do not have the complete topological information and are only aware of their neighboring nodes. Hence, sensor nodes cannot form a spanning tree.

The nodes that have the ability to directly communicate with each other are considered as neighbors. We also assume that the sensor nodes are unaware of the measurement model parameters of the other nodes. In a special case of high dimensional measurements, we relax this condition and allow each sensor node to know all the measurement models. Consequently, sensor nodes cannot exchange raw measurements and/or relay raw measurements to a centralized fusion node; in the former case due to the unawareness of the measurement models, and in the latter case due to the high communication cost associated with it.

1.2 Motivation

Wireless sensor networks (WSNs) consist of resource constrained and battery-powered devices called sensor nodes. The sensor nodes have sensing, processing and communication capabilities. In many practical applications like surveillance [1] and habitat monitoring [2], these sensor nodes are deployed in a remote region and are not accessible. In order for wireless sensor networks to have a long lifetime, the limited battery power of the sensor nodes must be used efficiently. Most of the battery power is consumed by wireless transmissions and an efficient communication strategy can greatly enhance the network lifetime.

Distributed state estimation in WSNs is attractive due to many reasons. It does not have a single point of failure and allows any sensor node to be queried to retrieve the state estimate. It also eliminates the need for a centralized entity to have complete knowledge of the sensor model parameters. In the case of high dimensional measurements, local processing of data also reduces the amount of information that has to be transmitted within the network. The distributed estimation schemes are designed such that they minimize the communication overhead and increase the network lifetime.

The distributed Kalman filtering schemes [3, 4] have a low computational and communication overhead and perform well for linear and Gaussian problems. However, these methods have a poor performance for problems with non-linear state dynamics and/or non-linear measurement models. Distributed particle filtering methods [5, 6, 7, 8, 9] are attractive for non-linear estimation problems because they have the ability to approximate more general distributions. However, they have a much higher computational and communication cost.

Motivated by this fact, we strive to develop energy efficient distributed estimation schemes based on the ensemble Kalman filter (EnKF) [10]. The EnKF lies in between the Kalman filter and the particle filter. It uses a sample based approach to approximate the first two

moments of the posterior distribution. The EnKF propagates the samples using the state dynamics which allows it to capture the non-linearity in the state dynamics. The fact that the EnKF only approximates the first two moments of the posterior distribution allows it to work with a relatively small sample size compared to the particle filters. This makes the EnKF a good candidate for development of energy efficient distributed estimation schemes for non-linear and non-Gaussian problems.

The proposed distributed schemes identify the sufficient statistics and use gossip algorithm to produce state estimates locally at every node. We have used the randomized gossip algorithm [11] to perform gossip since most of the existing distributed filtering schemes have reported results based on this algorithm. In future work, one could investigate how different accelerated gossip procedures affect the proposed schemes.

1.3 Thesis Contribution and Organization

In Chapter 2, we provide the background information about the EnKF and review the existing distributed estimation schemes. First, the structure, algorithmic implementation, and variations and extensions of the EnKF are discussed. Following this, we provide a review of the distributed data aggregation schemes used in WSNs. Finally, we summarize the previous work on distributed estimation in the WSNs.

In Chapter 3, we propose distributed approximations of the three forms of the EnKF. In order to develop a distributed update mechanism we express the update equations in the EnKF in an alternative information form. We employ randomized gossip algorithm to reach consensus on the statistics that are required to perform local updates. We also apply three measurement model linearization strategies to the EnKF and developed the distributed approximations using each of these strategies.

Chapter 4 provides a simulation based performance comparison of the proposed distributed schemes and the existing schemes. First, we compare the filtering methods in the centralized setting to study their computation cost and explore the best linearization strategy for the EnKFs. Finally, we compare the proposed distributed schemes to the state-of-the-art distributed filtering schemes and study their communication efficiency.

In Chapter 5, we provide a summary of the work and present avenues for future work.

Chapter 2

Literature Review

This chapter reviews the structure of the ensemble Kalman filter, its implementation, and variations and extensions. It also provides a review of the state-of-the-art distributed filtering schemes along with an introduction to the decentralized computation methods employed by these schemes. Section 2.1 briefly outlines the KF and EKF leading to the formulation of the EnKF. Section 2.2 provides a detailed discussion on the previous research that has been carried out on the EnKF. In Section 2.3, an introduction to the decentralized computation schemes in the wireless sensor networks is presented. Section 2.4 reviews the previous state-of-the-art distributed estimation schemes.

2.1 Sequential State Estimation

In the sequential estimation problem, the goal is to estimate the system state at each time step based on the previous state and the recorded measurements. Details of several state estimation methods are discussed in subsequent sections. First, the formulation of the state space model is presented.

2.1.1 Overview

The system state is modeled by a discrete-time Markov process $\{\mathbf{x}_k, k \in \mathbb{N}\}$, where $\mathbf{x}_k \in \mathcal{R}^n$ represents the system state vector at time instant k. The time evolution of the system state

can be characterized by the equation

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}) + \mathbf{n}_{k-1}, \tag{2.1}$$

where $f : \mathcal{R}^n \rightarrow \mathcal{R}^n$ is a system model function (possibly non-linear) and $\mathbf{n}_{k-1}$ is the unknown system noise over one time step from $k-1$ to k. The state process $\{\mathbf{x}_k, k \in \mathbb{N}\}$ is not observable and needs to be estimated.

However, we can observe the measurement process $\{\mathbf{y}_k, k \in \mathbb{N}\}$, where $\mathbf{y}_k \in \mathcal{R}^m$ represents the available measurements at time instant k. The measurement model is characterized by the equation

$$\mathbf{y}_k = h(\mathbf{x}_k) + \epsilon_k, \tag{2.2}$$

where $h : \mathcal{R}^n \rightarrow \mathcal{R}^m$ is a measurement model function (possibly non-linear) and ϵ_k is the measurement noise.

The objective here is to obtain a sequential estimate of the posterior distribution $p(\{\mathbf{x}_0, .., \mathbf{x}_k\}|\{\mathbf{y}_1, .., \mathbf{y}_k\})$ or $p(\mathbf{x}_k|\{\mathbf{y}_1, .., \mathbf{y}_k\})$ at each time step k based on the transition density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, prior density $p(\mathbf{x}_0)$ and observations $\{\mathbf{y}_1, .., \mathbf{y}_k\}$.

In order to simplify the analysis the following assumptions are made:

- The process noise sequence $\{\mathbf{n}_k, k \in \mathbb{N}\}$ is zero mean and i.i.d.
- The measurement noise sequence $\{\epsilon_k, k \in \mathbb{N}\}$ is zero mean and i.i.d.
- The process noise $\mathbf{n}_i$ and measurement noise ϵ_j are uncorrelated ($\mathbb{E}\{\mathbf{n}_i \epsilon_j^T\} = 0 \ \forall i, j \in \mathbb{N}$).

2.1.2 Kalman Filter

The Kalman filter (KF) [12, 13] is a variance-minimizing algorithm with the assumption that the posterior distribution $p(\mathbf{x}_k|\{\mathbf{y}_1, .., \mathbf{y}_k\})$ is Gaussian at each time step. The Gaussian assumption allows the posterior distribution to be completely parameterized by its mean and covariance. If the model operator $f(\cdot)$ and measurement operator $h(\cdot)$ are known linear functions and the process noise, measurement noise and prior density are drawn from known Gaussian distributions then the posterior distribution $p(\mathbf{x}_k|\{\mathbf{y}_1, .., \mathbf{y}_k\})$ can be proved to be Gaussian [14].

The KF uses a two step process to produce an estimate of the system state. First, a forecast step is performed in which an estimate of the system state is produced by integrating the previous state forward in time using the system model, equation (3.1). In order to completely specify the pdf of the state forecast the known error covariance of the previous state is also evolved in time to produce the error covariance of the state forecast. Then, an analysis step is performed in which the observations are combined with the state forecast using a weighted linear combination. The weights are computed based on the error covariance of the state forecast and the error covariance of the measurement noise. This produces a refined state estimate along with its error covariance which is a more accurate representation of the true state of the system. We refer to this refined state estimate as the analyzed state estimate throughout the thesis.

Representation of error statistics

In the KF, the error covariance in the state forecast, $\mathbf{P}^f$, and the error covariance in analyzed state, $\mathbf{P}^a$, are defined as follows:

$$\mathbf{P}^f = \mathbb{E}\{(\mathbf{x}^f - \mathbf{x}^t)(\mathbf{x}^f - \mathbf{x}^t)^T\}, \tag{2.3}$$

$$\mathbf{P}^a = \mathbb{E}\{(\mathbf{x}^a - \mathbf{x}^t)(\mathbf{x}^a - \mathbf{x}^t)^T\}, \tag{2.4}$$

where vector $\mathbf{x}$ represents the system state and the superscripts f, a and t denote the forecast, analyzed and true states, respectively. We use the same convention throughout the thesis.

The forecast step

The forecast step uses the system model, equation (3.1). The forecast equation for the system state at time step k is given by

$$\mathbf{x}_k^f = \mathbf{F}\mathbf{x}_{k-1}^a, \tag{2.5}$$

where the system model function $f(\cdot)$ is replaced by a known matrix $\mathbf{F}$ that defines this linear function.

The error covariance evolves according to the equation

$$\mathbf{P}_k^f = \mathbf{F}\mathbf{P}_{k-1}^a\mathbf{F}^T + \mathbf{Q}, \tag{2.6}$$

where $\mathbf{Q}$ is the covariance matrix for the process noise. First, a forecast is made by using the initial state $\mathbf{x}_0^a$ and initial covariance matrix $\mathbf{P}_0^a$ which are defined by the prior density $p(\mathbf{x}_0)$.

The analysis step

In the KF, the measurement function $h(\cdot)$ is assumed to be linear, therefore the measurement model can be written as

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \epsilon_k, \tag{2.7}$$

where the measurement function $h(\cdot)$ is replaced by a known matrix $\mathbf{H}$ that defines this linear function.

The Kalman filter finds the optimal solution $\mathbf{x}_k^a$ such that the variance of the posterior distribution is minimized. This optimal solution minimizes the cost function

$$\mathcal{J}(\mathbf{x}_k^a) = (\mathbf{x}_k^f - \mathbf{x}_k^a)^T(\mathbf{P}_k^f)^{-1}(\mathbf{x}_k^f - \mathbf{x}_k^a) + (\mathbf{y}_k - \mathbf{H}_k\mathbf{x}_k^a)^T(\mathbf{R})^{-1}(\mathbf{y}_k - \mathbf{H}_k\mathbf{x}_k^a), \tag{2.8}$$

where $\mathbf{R}$ is the measurement noise covariance matrix.
The analyzed state $\mathbf{x}_k^a$ that minimizes the cost function defined above is given by

$$\mathbf{x}_k^a = \mathbf{x}_k^f + \mathbf{P}_k^f\mathbf{H}^T(\mathbf{H}\mathbf{P}_k^f\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{y}_k - \mathbf{H}\mathbf{x}^f). \tag{2.9}$$

The corresponding analyzed error covariance matrix is given by

$$\mathbf{P}_k^a = \mathbf{P}_k^f - \mathbf{P}_k^f\mathbf{H}^T(\mathbf{H}\mathbf{P}_k^f\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}\mathbf{P}^f. \tag{2.10}$$

These update equations can be succinctly expressed using the Kalman gain matrix, $\mathbf{K}_k$, as follows

$$\mathbf{x}_k^a = \mathbf{x}_k^f + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}\mathbf{x}^f), \tag{2.11}$$

$$\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^f, \tag{2.12}$$

where the Kalman gain matrix, $\mathbf{K}_k$ is given by

$$\mathbf{K}_k = \mathbf{P}_k^f\mathbf{H}^T(\mathbf{H}\mathbf{P}_k^f\mathbf{H}^T + \mathbf{R})^{-1}. \tag{2.13}$$

2.1.3 Extended Kalman Filter

The extended Kalman filter (EKF) [15] uses a linearized approximation of the non-linear system model and/or the non-linear measurement model. The forecast equations of the EKF are given by

$$\mathbf{x}_k^f = f(\mathbf{x}_{k-1}^a), \tag{2.14}$$

$$\mathbf{P}_k^f = \mathbf{F}'_{k-1}\mathbf{P}_{k-1}^a\mathbf{F}'^T_{k-1} + \mathbf{Q}. \tag{2.15}$$

where $\mathbf{F}'_{k-1}$ is the Jacobian or tangent linear operator of the the model function evaluated at $\mathbf{x}_{k-1}^a$ and can be written as

$$\mathbf{F}'_{k-1} = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\Big|_{\mathbf{x}_{k-1}^a}. \tag{2.16}$$

A similar procedure is used to obtain a linearized approximation of the measurement model. The linearization is performed at the most recent forecast and is given by

$$\mathbf{H}'_k = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}\Big|_{\mathbf{x}_k^f}. \tag{2.17}$$

The analysis equations for the mean and error covariance are as follows

$$\mathbf{x}_k^a = \mathbf{x}_k^f + \mathbf{K}_k(\mathbf{y}_k - h(\mathbf{x}_k^f)), \tag{2.18}$$

$$\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k\mathbf{H}'_k)\mathbf{P}_k^f. \tag{2.19}$$

Here, the Kalman gain matrix $\mathbf{K}_k$ is given by

$$\mathbf{K}_k = \mathbf{P}_k^f\mathbf{H}'^T_k(\mathbf{H}'_k\mathbf{P}_k^f\mathbf{H}'^T_k + \mathbf{R})^{-1}. \tag{2.20}$$

2.1.4 Ensemble Kalman Filter (EnKF)

The ensemble Kalman filter (EnKF) is a Monte Carlo (MC) scheme which was introduced by Evensen [10]. A review article by Evensen [16] provides detailed discussion on theoretical and practical aspects of the EnKF and serves as a reference document. The EnKF was designed to overcome the limitations of the KF and EKF in non-linear dynamical models and high-dimensional state spaces. The key idea in the EnKF is to represent the system state using an ensemble of model realizations. Each ensemble member is then evolved in time independently using the non-linear dynamical model. The mean of the ensemble members is considered as the state estimate and the deviation of ensemble members from their mean is used to define the error covariance. The analysis scheme also updates each ensemble member separately which produces the analyzed state ensemble.

In contrast to the KF, this approach allows state predictions to capture the non-linearity in the model. Although the EKF also considers this non-linearity by using a linearized approximation, its performance degrades severely in highly non-linear models. The linearized approximation in the EKF can also lead to unbounded linear instabilities [17]. Moreover, the KF and EKF require an error covariance matrix to be maintained and evolved in time, which becomes computationally expensive in high-dimensional state spaces. In the EnKF, the computational cost of maintaining and evolving the ensemble is limited by the fact that a relatively small ensemble size can yield good results. Therefore, the EnKF offers an alternative approach that has better performance and higher computational efficiency.

Representation of error statistics

In the EnKF, the system state is represented by an ensemble of state vectors $\{\mathbf{x}_{(i)}\}_{i=1}^{N}$. The ensemble mean is defined as the average of ensemble members

$$\hat{\mathbf{x}}^f = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_{(i)}^f, \tag{2.21}$$

$$\hat{\mathbf{x}}^a = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_{(i)}^a. \tag{2.22}$$

The error covariances are approximated by the sample covariances and are given by

$$\mathbf{P}^f = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}^f_{(i)} - \hat{\mathbf{x}}^f)(\mathbf{x}^f_{(i)} - \hat{\mathbf{x}}^f)^T, \tag{2.23}$$

$$\mathbf{P}^a = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}^a_{(i)} - \hat{\mathbf{x}}^a)(\mathbf{x}^a_{(i)} - \hat{\mathbf{x}}^a)^T. \tag{2.24}$$

The forecast step

The forecast equation in the EnKF has the same form as the forecast equation in the KF. In the EnKF, each ensemble member is updated independently using the full non-linear model function and an additional error term is added. The forecast equation for ensemble member $\mathbf{x}_{(i)}$ at time instant k is given by

$$\mathbf{x}^f_{(i)k} = f(\mathbf{x}^a_{(i)k-1}) + \mathbf{n}_{(i)k-1} \tag{2.25}$$

where the noise term is generated from a Gaussian distribution which has mean zero and covariance $\mathbf{Q}$. Hence, no explicit forecast of the the error covariance is needed. The forecast for the error covariance can be estimated from the predicted ensemble using equation (2.23).

The analysis step

In the analysis step all the quantities have the same time index so this index will be dropped from here onwards. The analysis step in the EnKF operates on each ensemble member separately. Burgers et al. [18] have shown that it is necessary to treat the observation as a random quantity. The random observations must have a Gaussian distribution with mean equal to the actual measurements and covariance equal to $\mathbf{R}$. Therefore, an ensemble of observations $\{\mathbf{y}_{(i)}\}_{i=1}^N$ is defined as follows

$$\mathbf{y}_{(i)} = \mathbf{y} + \epsilon_{(i)},\ i = 1, .., N, \tag{2.26}$$

where $\mathbf{y}$ is the recorded measurement at a particular time step and $\epsilon_{(i)}$ is generated from a Gaussian distribution with mean zero and covariance $\mathbf{R}$.

The analysis step performs the following update on each ensemble member $\mathbf{x}_{(i)}$

$$\mathbf{x}^a_{(i)} = \mathbf{x}^f_{(i)} + \mathbf{P}^f\mathbf{H}^T(\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{y}_{(i)} - \mathbf{H}\mathbf{x}^f_{(i)}), \tag{2.27}$$

The mean of the analyzed ensemble is considered as the best estimate. This can be expressed using the Kalman gain as follows

$$\hat{\mathbf{x}}^a = \hat{\mathbf{x}}^f + \mathbf{K}(\hat{\mathbf{y}} - \mathbf{H}\hat{\mathbf{x}}^f), \tag{2.28}$$

where $\hat{\mathbf{y}} = \mathbf{y}$, the actual recorded measurements and the Kalman gain $\mathbf{K}$ is defined as

$$\mathbf{K} = \mathbf{P}^f\mathbf{H}^T(\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R})^{-1}. \tag{2.29}$$

The expression for the error covariance obtained from the analyzed ensemble can be derived using equation (2.24) as follows

$$\begin{aligned}
\mathbf{P}^a &= \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}^a_{(i)} - \hat{\mathbf{x}}^a)(\mathbf{x}^a_{(i)} - \hat{\mathbf{x}}^a)^T, \\
&= (\mathbf{I} - \mathbf{K}\mathbf{H})\left(\frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}^f_{(i)} - \hat{\mathbf{x}}^f)(\mathbf{x}^f_{(i)} - \hat{\mathbf{x}}^f)^T\right)(\mathbf{I} - \mathbf{K}\mathbf{H})^T \\
&\qquad + \mathbf{K}\left(\frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{y}_{(i)} - \hat{\mathbf{y}})(\mathbf{y}_{(i)} - \hat{\mathbf{y}})^T\right)\mathbf{K}^T, \\
&= (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}^f(\mathbf{I} - \mathbf{H}^T\mathbf{K}^T) + \mathbf{K}\mathbf{R}\mathbf{K}^T, \\
&= \mathbf{P}^f - \mathbf{K}\mathbf{H}\mathbf{P}^f - \mathbf{P}^f\mathbf{H}^T\mathbf{K}^T + \mathbf{K}(\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R})\mathbf{K}^T, \\
&= (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}^f.
\end{aligned} \tag{2.30}$$

It can be observed that the mean and the error covariance of the analyzed ensemble in the EnKF, equations (2.28) and (2.30), have exactly the same expressions as the analysis equations (2.11) and (2.12) in the KF. The only difference is that the EnKF uses the error covariances that are defined using an ensemble of state realizations. Therefore, in the limit of an infinite ensemble size the analysis performed in the EnKF will give exactly the same result as analysis in the KF. The complete EnKF filtering approach is presented in Algorithm 1.

In the EnKF, the system model is assumed to be non-linear, therefore the posterior distri-

Algorithm 1 Ensemble Kalman filter

// Initialization at time $k = 0$
for each sample $i = 1, 2, .., N$ **do**
  Sample: $\mathbf{x}^a_{(i)0} \sim p(\mathbf{x}_0)$
end for

// For time $k > 0$
for $k = 1, 2, .., T$ **do**

  // Forecast step
  for each sample $i = 1, 2, .., N$ **do**
    Sample: $\mathbf{n}_{(i)k-1} \sim \mathcal{N}(0, \mathbf{Q})$
    $\mathbf{x}^f_{(i)k} = f(\mathbf{x}^a_{(i)k-1}) + \mathbf{n}_{(i)k-1}$
  end for
  $\hat{\mathbf{x}}^f_k = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}^f_{(i)k}$
  $\mathbf{P}^f_k = \frac{1}{N-1} \sum_{i=1}^{N} (\mathbf{x}^f_{(i)k} - \hat{\mathbf{x}}^f_k)(\mathbf{x}^f_{(i)k} - \hat{\mathbf{x}}^f_k)^T$

  // Analysis step
  for each sample $i = 1, 2, .., N$ **do**
    Sample: $\epsilon_{(i)k} \sim \mathcal{N}(0, \mathbf{R})$
    $\mathbf{y}_{(i)k} = \mathbf{y}_k + \epsilon_{(i)k}$
    $\mathbf{K}_k = \mathbf{P}^f_k \mathbf{H}^T (\mathbf{H} \mathbf{P}^f_k \mathbf{H}^T + \mathbf{R})^{-1}$
    $\mathbf{x}^a_{(i)k} = \mathbf{x}^f_{(i)k} + \mathbf{K}_k (\mathbf{y}_{(i)k} - \mathbf{H} \mathbf{x}^f_{(i)k})$
  end for

end for

bution $p(\mathbf{x}_k | \{\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_k\})$ will have non-Gaussian contributions and will not be parametrized with mean and covariance only. Hence, the analysis performed in the EnKF is an approximation using the mean and covariance of the posterior distribution. On the other hand, the ensemble will capture non-Gaussian components during the forecast, which is performed using the non-linear model function. A more sophisticated analysis scheme can be used that performs the Bayesian update, which leads to the particle filtering theory [19]. However, particle filtering methods have a much larger computational cost that makes them infeasible for high dimensional state spaces or computationally constrained systems.

2.1.5 Ensemble Square Root Filter (ESRF)

The ensemble square root filter (ESRF) [20] is a variant of the EnKF and only differs in the way the analysis step is performed. This scheme calculates a transformation matrix which when left-multiplied with a state forecast sample produces the updated sample. The ESRF uses the KF analysis equation (2.11) to compute the mean of the analyzed ensemble using the mean of the forecast ensemble. The deviation of the updated ensemble members from their mean is then obtained by applying a linear transformation to the deviation of the forecast ensemble members from their mean. The deviation of a sample from its mean is calculated as $\mathbf{x}^{'} = \mathbf{x} - \hat{\mathbf{x}}$, where $\hat{\mathbf{x}}$ is the sample average. The proposed transformation operates as follows

$$\mathbf{x}^{'a} = \mathbf{T}\mathbf{x}^{'f}, \tag{2.31}$$

where $\mathbf{T}$ is given by

$$\mathbf{T} = (\mathbf{I} - \mathbf{K}\mathbf{H})^{1/2}, \tag{2.32}$$

where the $(\cdot)^{1/2}$ denotes the unique positive definite square root of a positive definite matrix. This approach is described in Algorithm 2.

2.1.6 Deterministic Ensemble Kalman Filter (DEnKF)

In the deterministic ensemble Kalman filter (DEnKF) [21], the analysis scheme is similar to the ESRF and it also uses a transformation matrix to obtain the updated ensemble from the forecast ensemble. This scheme does not need to compute a matrix square root in order to evaluate the transformation matrix, hence it has a lower computational cost as compared to the ESRF. In the DEnKF, the transformation matrix is defined as

$$\mathbf{T} = \mathbf{I} - \frac{1}{2}\mathbf{K}\mathbf{H}. \tag{2.33}$$

The DEnKF filtering algorithm is presented in Algorithm 3.

Algorithm 2 Ensemble square root filter

// Initialization at time $k = 0$
for each sample $i = 1, 2, .., N$ **do**
  Sample: $\mathbf{x}^a_{(i)0} \sim p(\mathbf{x}_0)$
end for

// For time $k > 0$
for $k = 1, 2, .., T$ **do**

  // Forecast step
  for each sample $i = 1, 2, .., N$ **do**
    Sample: $\mathbf{n}_{(i)k-1} \sim \mathcal{N}(0, \mathbf{Q})$
    $\mathbf{x}^f_{(i)k} = f(\mathbf{x}^a_{(i)k-1}) + \mathbf{n}_{(i)k-1}$
  end for
  $\hat{\mathbf{x}}^f_k = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}^f_{(i)k}$
  $\mathbf{P}^f_k = \frac{1}{N-1} \sum_{i=1}^{N} (\mathbf{x}^f_{(i)k} - \hat{\mathbf{x}}^f_k)(\mathbf{x}^f_{(i)k} - \hat{\mathbf{x}}^f_k)^T$

  // Analysis step
  $\mathbf{K}_k = \mathbf{P}^f_k \mathbf{H}^T (\mathbf{H}\mathbf{P}^f_k \mathbf{H}^T + \mathbf{R})^{-1}$
  $\hat{\mathbf{x}}^a_k = \hat{\mathbf{x}}^f_k + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}\hat{\mathbf{x}}^f_k)$
  $\mathbf{T}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})^{1/2}$
  for each sample $i = 1, 2, .., N$ **do**
    $\mathbf{x}^a_{(i)k} = \hat{\mathbf{x}}^a_k + \mathbf{T}_k(\mathbf{x}^f_{(i)k} - \hat{\mathbf{x}}^f_k)$
  end for

end for

2.2 Previous Research on EnKF

2.2.1 Development of EnKF

The ensemble Kalman filter (EnKF) was introduced by Evensen in [10]. In this paper, a Monte Carlo (MC) technique is presented as an alternative to solving the error covariance equation in the EKF. In this approach, an ensemble of state realizations is propagated in time and the error covariance is calculated using the ensemble. It is concluded that the EnKF eliminates the unbounded error growth suffered by the EKF because of its approximate

Algorithm 3 Deterministic ensemble Kalman filter

1: `// Initialization at time` $k = 0$
2: **for** each sample $i = 1, 2, .., N$ **do**
3: Sample: $\mathbf{x}^a_{(i)0} \sim p(\mathbf{x}_0)$
4: **end for**
5: `// For time` $k > 0$
6: **for** $k = 1, 2, .., T$ **do**
7: `// Forecast step`
8: **for** each sample $i = 1, 2, .., N$ **do**
9: Sample: $\mathbf{n}_{(i)k-1} \sim \mathcal{N}(0, \mathbf{Q})$
10: $\mathbf{x}^f_{(i)k} = f(\mathbf{x}^a_{(i)k-1}) + \mathbf{n}_{(i)k-1}$
11: **end for**
12: $\hat{\mathbf{x}}^f_k = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}^f_{(i)k}$
13: $\mathbf{P}^f_k = \frac{1}{N-1}\sum_{i=1}^{N} (\mathbf{x}^f_{(i)k} - \hat{\mathbf{x}}^f_k)(\mathbf{x}^f_{(i)k} - \hat{\mathbf{x}}^f_k)^T$
14: `// Analysis step`
15: $\mathbf{K}_k = \mathbf{P}^f_k\mathbf{H}^T(\mathbf{H}\mathbf{P}^f_k\mathbf{H}^T + \mathbf{R})^{-1}$
16: $\hat{\mathbf{x}}^a_k = \hat{\mathbf{x}}^f_k + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}\hat{\mathbf{x}}^f_k)$
17: $\mathbf{T}_k = (\mathbf{I} - \frac{1}{2}\mathbf{K}_k\mathbf{H})$
18: **for** each sample $i = 1, 2, .., N$ **do**
19: $\mathbf{x}^a_{(i)k} = \hat{\mathbf{x}}^a_k + \mathbf{T}_k(\mathbf{x}^f_{(i)k} - \hat{\mathbf{x}}^f_k)$
20: **end for**
21: **end for**

closure scheme. It is also demonstrated that the EnKF can provide a good estimate using an ensemble of moderate size by applying it to a non-linear quasi-geostrophic model.

In [18], Burgers et al. reexamine the analysis scheme used by the EnKF. They point out that the derivation of the EnKF assumes that the measurements are random variables and violation of this assumption leads to an ensemble with too low variance. They develop a new analysis scheme in which random perturbations are added to the measurements at each analysis step to generate an ensemble of measurements. These perturbations are generated from a distribution with zero mean and the covariance equal to the measurement error

covariance.

In [22], Evensen et al. derive a general filter that solves the Bayesian problem of estimating the posterior distribution from the predicted pdf and observations. The EnKF is then re-derived as a special case of the general filter which provides a sub-optimal solution assuming the prior density is Gaussian.

In a review paper [16], an extensive discussion on theoretical and practical aspects of the EnKF is provided. This paper provides a summary of all the previous research on the EnKF along with a new interpretation of the EnKF. It is emphasized that the interpretation that analysis in the EnKF is sought in the space spanned by ensemble members allows for efficient algorithms to be developed. The algorithmic formulation of the EnKF is also discussed along with an efficient numerical implementation of the EnKF.

The ensemble Kalman filter for combined parameter and state estimation is derived in [23]. The problem is formulated as estimating the joint probability distribution of the model parameters and the state variables given the measurements. The implementation approach uses the prior density for the poorly known parameters to generate an ensemble of parameter realizations. This ensemble of parameter realizations is concatenated with the state ensemble and updated at each time step using the variance minimizing analysis scheme.

2.2.2 Square Root Filters

The analysis scheme in the EnKF uses perturbation of measurements to generate an ensemble of measurements. The perturbation of measurements introduces an additional sampling error which is negligible in most cases. However, this sampling error becomes substantial when the number of measurements is larger than the ensemble size and considerably degrades the performance of the EnKF [24]. For these situations, analysis schemes have been developed that do not use perturbation of the measurements; these are referred to as square root implementations of the EnKF. In this section several square root implementations of the EnKF are summarized.

Bishop et al. [25] devise an implementation of the EnKF named the ensemble transform Kalman filter (ETKF). In this approach transformations of the state ensemble are defined that expedite the evaluation of forecast and analyzed error covariances. Instead of propagating every ensemble member forward in time a single ensemble forecast is performed followed by transformations to obtain the forecast ensemble. Similarly, the analysis scheme also ap-

plies a transformation on the forecast ensemble that results in the analyzed ensemble having mean and covariance consistent with the KF. This scheme is used by the authors to rapidly estimate the effect of the configuration of observations on the predicted error covariance and identify crucial points for sensor deployment.

Anderson et al. [26] present a related method named the ensemble adjustment Kalman filter (EAKF). In this scheme a linear operator is computed which when applied to the forecast ensemble generates the analyzed ensemble. It is shown that this operator always exists and can be computed efficiently. A comparison is carried out between the EAKF and the traditional EnKF using a small number of ensemble members and results show that the EAKF gives a better estimate of analyzed state covariance.

In [27], Whitaker et al. present another square root implementation of the EnKF named the ensemble square root filter (EnSRF). The analysis scheme presented here is similar in form to analysis in the EnKF, but differs in two ways. In the EnSRF the perturbation of the measurements is avoided and the Kalman gain in the analysis equation is replaced with a reduced Kalman gain. It is shown that perturbation of the measurement in the EnKF analysis prevents it from underestimating the analyzed error covariance. The use of a reduced Kalman gain is presented as an alternative way of preventing the underestimation in the error covariance. The results show that the EnSRF gives more accurate prediction for error covariances when the ensemble size is small (10-20 members).

Tippett et al. [28] point out that the deterministic transformation from the forecast ensemble to the analyzed ensemble is not unique and this non-uniqueness can be exploited to design filters with suitable properties. They provide a discussion on the previously mentioned square root filters [25, 26, 27] and compare their computational and numerical properties.

Evensen [29] develops an implementation of the EnKF that uses a deterministic analysis scheme. In this approach the mean of the analyzed ensemble is calculated from the standard KF analysis equation. The deviation of the analyzed ensemble members from their mean is then expressed as a linear transformation of the deviation of the forecast ensemble members from their mean. The analysis scheme developed here suffers from several drawbacks. Wang et al. [30] pointed out the introduction of bias in the mean, and in [31] Leeuwenburgh et al. demonstrate that most of the error covariance is represented by a few outliers in the ensemble.

Livings et al. [32] provide theoretical discussion on square root filters. They analyze the linear transformation that maps forecast ensemble perturbations (deviation of the ensemble

members from their mean) to analyzed ensemble perturbations. They provide sufficient conditions for the transformation matrix under which analysis is unbiased. It is shown that the transformation matrix is unbiased, meaning it preserves the zero mean of ensemble perturbations, if the vector of all ones is an eigenvector of the transformation matrix. This condition is satisfied if the transformation matrix is symmetric. It is further shown that if a new transformation is defined by post-multiplying an unbiased transformation matrix with another matrix which has the vector of all ones as its eigenvector, the new transformation is also unbiased.

In [20], Sakov et al. provide similar discussion on theoretical aspects of square root filters. They show that a mean preserving transformation can be represented as a product of a symmetric matrix and an orthonormal mean preserving matrix. They use this property to devise a transformation that results in correct error covariance while preserving the mean of the ensemble.

Evensen [17] builds on [20] and presents a symmetrical square root analysis scheme which uses a symmetric transformation matrix. In order to enable the ensemble perturbations to better resemble a Gaussian distribution a randomization in the analysis scheme is suggested. In this case, a new transformation is defined by post-multiplying a symmetric transformation matrix by a random orthogonal mean preserving matrix which can be computed using the algorithm from [20].

In [33], Reich et al. propose a transformation scheme that does not have any prior assumptions about the form of the prior and posterior distributions. This approach finds an optimal transportation [34] between the forecast state variables and the analyzed state variables. The optimal coupling between the forecast and the analyzed state variables is found by using linear programming [35]. The optimization metric is the squared Euclidean distance between the forecast and the analyzed state variables, which needs to be minimized. The analyzed ensemble is computed using a deterministic transformation based on the optimal coupling. In a review paper [36], Cotter et al. provide a discussion on the use of the optimal transportation and variable coupling for state estimation along with the most recent advances in this area.

2.2.3 Ensemble based Kalman-Bucy Filters

The Kalman-Bucy filter deals with continuous time dynamical systems. The ensemble based extension of the Kalman-Bucy filter [13] was introduced by Bergemann et al. [37]. In this approach, the analysis step is formulated in terms of ordinary differential equations (ODEs). The forecast ensemble perturbations (deviation of the ensemble members from their mean) are used as initial conditions and the solution of these ODEs over one time step gives the analyzed ensemble perturbations. The mean of the analyzed ensemble is computed from the standard KF analysis equation and used to obtain the analyzed ensemble. In [38], Bergemann et al. present another formulation of the analysis step using ODEs. In this scheme, the analyzed ensemble is directly obtained as a solution of the ODEs over a time step. Amezcua et al. [39] provide a discussion on the above approaches [37, 38]. They point out that in both of these approaches the stability of the filter depends on the ratio of the forecast error covariance to the observation error covariance. It is shown that if the time interval between two consecutive analysis steps is large, the magnitude of this ratio is also large which makes the filter unstable. In order to improve the stability of the filter a modification to these approaches is presented. The analyzed ensemble perturbations are represented as a product of the forecast ensemble perturbations and a matrix of weights. The weights are then computed at each analysis step as a solution of the ODEs. This results in reducing the dimension of the variables for the analysis step and improves the stability of the filter.

2.2.4 EnKF with Gaussian Mixture Models

The ensemble Kalman filter assumes that the prior distribution and model errors are Gaussian which lead to a Gaussian posterior distribution. Therefore, the posterior distribution can be characterized by mean and covariance only. In some applications the prior distribution cannot be accurately modeled by a Gaussian distribution. An example of such a situation is the reservoir models where the prior distribution can be multimodal [40]. Furthermore, even if the prior distribution is accurately modeled by a Gaussian distribution, strong non-linearities in the dynamical model can give rise to significant non-Gaussian components in the posterior distribution. In both these situations, modeling of the posterior distribution by a Gaussian distribution will result in an inaccurate estimate which limits the use of the EnKF. Several methods have been developed that overcome this problem by

modeling the prior and posterior distribution using Gaussian mixture models (GMMs). A Gaussian mixture distribution consists of a weighted sum of Gaussian distributions and has the ability to represent more general probability distributions.

In [41], Dovera et al. have extended the EnKF to incorporate multimodal distributions. They model the multimodal prior distributions using a Gaussian mixture. They use the results from [42] which show that if the prior is a Gaussian mixture, the system and measurement models are linear and the likelihood is a Gaussian, the posterior distribution is also a Gaussian mixture. At each analysis step the forecast distribution is assumed to be a Gaussian mixture and the parameters of this distribution are learned from the forecast ensemble using the expectation-maximization (EM) algorithm [43]. The number of components in the Gaussian mixture is assumed to be known beforehand as prior information. The EM algorithm is used to learn the mean and covariance of each component along with the membership probabilities of the ensemble members. The membership probability of an ensemble member is the probability that the member belongs to a particular component of the Gaussian mixture. Then for each ensemble member a random component index is selected based on the membership probabilities. Each ensemble member is associated with the component selected in the previous step and updated using the mean and covariance of this component. This generates samples from the posterior Gaussian mixture distribution.

Frei et al. [44] develop a similar scheme in which they use the Gaussian mixture distributions to model the prior and posterior distributions. The authors present a two stage process to perform state estimation. In the first stage, the forecast ensemble is divided into clusters in such a way that each ensemble member can be associated with more than one cluster. The clustering algorithm also determines the association probabilities between the ensemble members and the clusters. Each cluster is considered as a component and the component mean and covariance are computed using the association probabilities. In the second stage, the updated weight of each mixture component is computed based on the observations. Then samples are generated by randomly selecting a component using new weights and sampling from this component. In this sampling procedure each sample is associated with a single component. The analyzed ensemble members are then obtained from these samples by applying the EnKF analysis equations by using the mean and covariance of the corresponding component.

In [45], Stordal et al. identify a connection between the EnKF and the Gaussian mixture filter [46, 47]. The EnKF is viewed as a special case of Gaussian mixture filter with the added

constraint that all the weights are uniform. In this case, each ensemble member is treated as a mixture component. The authors present a new scheme named the adaptive Gaussian mixture filter (AGM) which relaxes this constraint. They introduce a weight interpolation procedure for the forecast ensemble in the Gaussian mixture filter which is given by

$$w_{\alpha}^{i} = \alpha w^{i} + (1 - \alpha)N^{-1}, \tag{2.34}$$

where α is the tuning parameter and determines how much the original weights are interpolated towards uniform weights. A performance comparison of the AGM with the EnKF is performed for reservoir models in [48]. The results show that the AGM estimates have higher correlation with the reference data compared to the EnKF. Rezaie et al. [49] develop a shrinkage scheme in the Gaussian mixture filter that has the same effect as weight interpolation in AGM. A tuning parameter is introduced that is used to perform the shrinkage of the forecast ensemble. Each shrunken ensemble member is generated by taking a weighted average of the ensemble member with the ensemble mean. The results indicate that the proposed method has better performance compared to the EnKF.

In [50], Reich incorporates the Gaussian mixture prior and posterior distributions in a continuous formulation of the EnKF [38]. Initially, the expectation maximization algorithm is used to determine the parameters of the Gaussian mixture distribution from the forecast ensemble. A heuristic approach is used to decide on the number of components in the Gaussian mixture. The analyzed ensemble is obtained as a numerical solution of the ordinary differential equations over a time step. The results show that this scheme gives better estimates when the ensemble distribution is non-Gaussian and/or multimodal.

2.2.5 Localization

The EnKF is a Monte Carlo technique that uses an ensemble of state realizations to perform estimation. The use of finite sized ensemble leads to undesirable sampling errors like spurious correlations between uncorrelated variables. These spurious correlations result in measurements that are actually uncorrelated with the state variables having a small impact on the state ensemble during the analysis phase. This causes a reduction in the variance of the state ensemble and the covariance computed from the state ensemble becomes an underestimate. The effect of spurious correlations is reduced by using a large ensemble. The reduction in the variance can also lead to filter divergence. In order to eradicate the effects

of spurious correlations several types of localization schemes are used in the EnKF. The localization schemes can be classified as covariance localization, local analysis and adaptive localization. These localization schemes are explained in subsequent sections.

Covariance Localization

Covariance localization is based on the premise that variables that are separated by a large spatial distance must have a very low correlated. Covariance localization was introduced by Houtekamer et al. [51]. In this approach the authors use a Schur product (element-wise multiplication) of the covariance matrix with a specified correlation matrix to eliminate long-range spurious correlations and perform spatial smoothing. The Schur product theorem [52] states that the Schur product between a covariance matrix and correlation matrix is also a covariance function. The correlation function is constructed in such a way that it has local support and decreases monotonically with distance. In [53] Hamill et al. use the same approach and perform a Schur product between the covariance matrix and a correlation function with local support. They use a distance dependent correlation function which is a fifth order function that resembles the Gaussian function [54]. This correlation function is more compactly supported than the Gaussian function and reduces to zero after a finite radius. They conclude that this distance dependent reduction of covariance reduces spurious correlations and the optimal design of the correlation function strongly depends on the size of the ensemble. Bergemann et al. [38] introduce covariance localization in the continuous formulation of the EnKF. They also employ the Schur product between the covariance and the correlation functions to eliminate spurious correlations.

Local analysis

Another way to reduce the impact of spurious correlations is to use local analysis. In local analysis a model grid is constructed and analysis is performed grid point by grid point. Only the measurements that are spatially close to a state variable are used to update that variable. This approach is also computationally efficient when a large number of measurements have to be used. In [16] an implementation of the local analysis scheme is provided. This approach first computes the global analysis and then in order to obtain the local analysis at a grid point a correction term is added to the global analysis. In [55] Ott et al. present another local analysis scheme called the local ensemble Kalman filter. In this approach they define

local regions and use the local state vectors which have a much lower dimension as compared to the global state vector. They perform analysis on each local low-dimensional subspaces and combine them to form the global ensemble. It is concluded that local analysis gives advantages in terms of eliminating spurious correlations and reducing the computation time.

Adaptive localization

Adaptive localization methods try to explore the need for localization and try to find the optimal localization strategy. This is extremely useful for dynamical models where the correlations and ensemble size changes over time. Anderson [56] presents a method that he names the hierarchical ensemble filter. In this hierarchical approach several ensembles are used to estimate the effect of spurious correlation between observations and state variables. An ensemble of ensemble Kalman filters is used to learn a localization function. In [57] Fertig et al. present a local analysis scheme for cases where observations are parameters of the state variables. Initially, the covariance matrix of the predicted observations is computed globally using all the state variables and used to estimate the correlations. The analysis is then performed locally by using only the observations that have considerable correlation with the state variables. A similar scheme is developed by Bishop et al. [58] where a moderation function is learned using the predicted covariance. The moderation function is constructed from powers of the ensemble covariance. The moderation function damps small correlations and eliminates the variance reduction caused by spurious correlations.

2.2.6 Inflation

Covariance inflation is another method to counter the variance reduction caused by spurious correlations. In this approach the deviation of the forecast ensemble members from their mean is slightly increased by multiplication with an inflation factor. This is given by the equation

$$\mathbf{x}_{(i)} = \rho(\mathbf{x}_{(i)} - \hat{\mathbf{x}}) + \hat{\mathbf{x}}, \tag{2.35}$$

where $\mathbf{x}_{(i)}$ is the i^{th} member of the forecast ensemble, $\hat{\mathbf{x}}$ is the mean of the forecast ensemble and ρ is the inflation factor. The inflation factor has a value that is slightly greater than one.

Several approaches have been proposed that adaptively estimate the optimal value of the inflation parameter. In [59], a method for online estimation of the inflation factor is presented. An inflation variable is augmented with the state ensemble and estimated as a parameter at each time step using the analysis scheme in the EnKF. Evensen [17] provides a scheme to approximate the inflation factor. In this approach an ensemble matrix is generated such that each row of this matrix has zero mean and unit variance. This ensemble matrix is then analyzed at each analysis step using the measurements. The standard deviation of each row of the analyzed ensemble matrix is computed and an average is calculated. The reciprocal of this average is considered to be an approximation of the inflation factor. In [60], a discussion on the online estimation of the inflation factor is provided. It is found that in order to get a good estimation of the inflation factor an accurate knowledge of measurement error statistics is required and vice versa. It is shown that simultaneous estimation of the inflation factor and measurement error statistics gives better results. Reich [61] extends the inflation technique to the continuous formulation of the EnKF [38]. An additional term is introduced in the continuous analysis equations that does not affect the mean of the ensemble but causes a slight increase in the ensemble deviations. The similarities between the proposed equations and the analysis equations in the continuous H_∞ filter [62] are also discussed.

2.2.7 Model Bias Estimation

Several schemes have been developed that use the EnKF to estimate the model bias or systematic model errors. Keppenne et al. [63] introduce a scheme for estimating the model bias using the EnKF. The idea is to add a new variable for model bias in the state vector. The variable is then estimated along with the state variables using the EnKF analysis scheme. It is shown that their approach has the ability to correct systematic errors in the model. Fertig et al. [64] develop a similar scheme based on augmenting the state vector with a bias variable. The online estimation of bias is done using variance minimizing analysis in the EnKF. They also concluded that the bias estimation helps in reducing the systematic model errors. In [65], Baek et al. present several approaches for estimating the model bias. They develop different models for parameterizing the bias. They show that in order to obtain good results the choice of bias parameterization becomes crucial.

2.2.8 Other Extensions and Convergence

In [66], Gottwald et al. propose a variance-limiting ensemble Kalman filter for situations where some of the variables in the observation space are not directly observable. It is assumed that some prior knowledge about the statistical behavior of these unobserved variables is available. In such a scenario, the analysis performed using the EnKF results in an overestimation of the error covariance. A variance-limiting scheme is devised in the EnKF that counteracts this overestimation of the error covariance.

Lorentzen et al. [67] present an iterative version of the EnKF. If the observation model is non-linear the analysis in the EnKF is not accurate and an iterative procedure is employed to improve the estimation accuracy by repeatedly updating the ensemble until the stopping condition is met. The results show that the proposed iterative filter gives more accurate estimates compared to the EnKF for non-linear observation models.

In [68], Mandel et al. provide a discussion on the convergence of the EnKF. They show that the EnKF in the limit of an infinite ensemble size converges to the Kalman filter. The weak law of large numbers for exchangeable random variables is used to prove the convergence of the ensemble based covariance in the EnKF to the covariance in the KF. It is further discussed that the rate of convergence of the EnKF is increased by using localization techniques.

2.2.9 Parallelization in EnKF

In the EnKF, the ensemble members are updated separately which makes it suitable for parallel implementation. Parallelizing the EnKF can be advantageous when dealing with a high dimensional model and can result in a large reduction in the computation time. In [69], Tavakoli et al. propose a parallelization scheme for both forecast and analysis steps in the EnKF. In the forecast step, the ensemble members are divided into groups which are passed to different processors. The forecast for each group is performed simultaneously and the result is communicated back to the central processor. In the analysis step, the central processor uses the observations and the forecast ensemble to calculate the Kalman gain. The Kalman gain is used to calculate a coefficient matrix, which when post-multiplied with a row of the forecast ensemble matrix yields the corresponding row of the analyzed ensemble matrix. The rows of the forecast ensemble matrix are then passed to different processors along with the coefficient matrix. The multiplication between a row of the forecast ensemble matrix and the

coefficient matrix is carried out locally and the result is passed back to the central processor. Xu et al. [70] present another parallelization scheme for the EnKF. In this approach, the ensemble is divided into partitions and each partition is sent to a different processor. The forecast step for each partition is carried out simultaneously at each processor. The analysis step is also parallelized and proceeds as follows. First, each processor computes a local estimate of the mean and the covariance using its forecast ensemble. These local estimates are communicated back to a central processor which uses them to compute the covariance and the Kalman gain. The value of the Kalman gain is sent to all the processors and they update their ensemble concurrently.

2.2.10 Discussion

The EnKF is a state estimation approach that was designed to overcome the weakness of the KF and EKF. Although the EnKF has been successfully applied in many state estimation problems, across different research areas, it does not provide good performance in all scenarios. Therefore, several problem specific modifications and extensions of the EnKF have been proposed to improve its performance. In this section, a discussion on the applicability of several extensions of the EnKF is provided.

If a large number of measurements are used then sampling errors due to perturbation of measurements become large and performance of the EnKF degrades. A considerable decrease in the estimation accuracy occurs when the number of measurements are larger than the ensemble size. These sampling errors can be reduced by increasing the ensemble size. An alternative approach is to avoid these sampling errors by using a square root implementation of the EnKF. In square root filters no perturbation of measurements is performed. Therefore, square root filters are a good choice when a large number of measurements are used while the ensemble size is relatively small.

The continuous extensions of the ensemble Kalman filter (ensemble Kalman-Bucy filter) have been developed for the problems that require frequent measurement analysis. If the measurements are available very frequently and need to be used immediately, the ensemble based Kalman-Bucy filters are a suitable choice.

Gaussian mixture models (GMMs) are used in the EnKF if the prior distribution is known to have a non-Gaussian and/or a multimodal structure or the model dynamics are highly non-linear. In both these cases, the posterior distribution develops significant non-Gaussian

components and cannot be represented by a Gaussian distribution. In these situations, modeling the prior and posterior distributions with a Gaussian mixture distribution gives better results in terms of estimation accuracy.

When the ensemble size in the EnKF is small, spurious correlations appear between uncorrelated variables. The spurious correlations result in the underestimation of the error covariance. Localization and covariance inflation are two techniques that counter the effect of spurious correlations. Both these techniques have a similar effect and either one of them can be used to avoid the underestimation of the error covariance. If the system dynamics change over time, meaning that the correlation between different variables changes over time then adaptive localization and inflation techniques are more suitable. The two localization approaches, namely, covariance localization and local updating, have comparable performance in terms of reducing spurious correlations. However, local updating has an added advantage of decreasing the computational cost.

2.3 Decentralized Computation in Wireless Sensor Networks

A wireless sensor network (WSN) consists of many battery-powered devices, known as sensor nodes, that have sensing, data processing and communication capabilities. The sensor nodes have limited resources, specifically the power supply and the communication bandwidth. The sensor nodes are only capable of short range communication, hence a sensor node cannot directly communicate with all the other nodes in the network. In order to aggregate information in these wireless networks specialized schemes are required.

Several approaches have been developed that perform data aggregation in wireless sensor networks (WSNs). The schemes [71, 72, 73] require specific routing services from the network, e.g. formation and maintenance of cyclic paths that pass through each node in the network. Other methods [74, 75, 76] construct and maintain spanning trees rooted at the sink node. In these schemes, the data reaches the sink node by propagating upwards through the tree. Both types of methods have several shortcomings. They have a single point of failure, introduce communication bottlenecks near the sink node and increase the communication overhead due to maintenance of routing paths. Gossip algorithms are decentralized methods for aggregating data in WSNs and do not suffer from the aforementioned drawbacks. In [77], an extensive survey on gossip algorithms is presented. In the subsequent section, we present the randomized gossip algorithm [11] that solves the *average consensus* problem in WSNs.

2.3.1 Randomized Gossip Algorithm

In the *average consensus* problem, every node $v \in V$(the set of nodes), starts with a local variable (vector or a matrix) $\zeta^v(0)$, and the goal is to compute the element-wise average of this variable over the network, $\overline{\zeta} = |V|^{-1} \sum_{v \in V} \zeta^v(0)$. The randomized gossip algorithm iteratively and asynchronously performs data exchanges between the neighboring nodes. The nodes that have the ability to directly communicate with each other are considered as neighbors. In iteration k, a pair of neighboring nodes (v, w) is selected randomly. These nodes exchange their local values and update their local estimates as $\zeta^v(k) = \zeta^w(k) = \frac{1}{2}(\zeta^v(k-1) + \zeta^w(k-1))$. Under mild conditions on the network connectivity, as $k \to \infty$, all the nodes posses a local estimate $\zeta^v(k)$, $v \in V$ that is equal to the true average $\overline{\zeta}(0)$. The communication cost of such a decentralized computation scheme can be measured in terms of the total number of scalars that are transmitted in the network. In each iteration of this algorithm both the selected nodes transmit their local variables to each other. Assuming that the local variable ζ^v contains M scalars and the algorithm runs for K iterations, the total number of scalars transmitted is $2MK$.

2.4 Distributed State Estimation

Distributed estimation techniques in wireless sensor networks are becoming increasingly popular. This can be attributed to the fact that these schemes distribute the communication load evenly in the network and do not have a single point of failure. In contrast to relaying the raw measurements to a sink, such schemes perform local processing and only transmit the essential information which significantly reduces the communication overhead when the measurement dimensionality is high. The local processing also eliminates the need for a central entity to have the complete knowledge of the sensor measurement models and allows retrieval of the state estimate from any sensor node. In the following sections, we present several schemes that perform distributed estimation in wireless sensor networks.

2.4.1 Distributed Kalman Filtering

The distributed Kalman filtering approaches [3, 4] identify and aggregate through the network, the statistics, that are sufficient to implement the KF analysis equations locally at every node. These methods provide an efficient solution to the linear estimation problem, i.e.,

Gaussian problem, but do not work well with non-linear state dynamics and/or non-linear measurement models. The schemes in [78, 79, 80] employ the unscented transformation to develop the distributed unscented Kalman filters (UKFs) that have the ability to capture the non-linearities in both the state dynamics and the measurement models. These distributed UKF schemes also aggregate the sufficient statistics through the network and perform the analysis locally.

2.4.2 Distributed Particle Filtering

Distributed particle filters have the ability to represent more general posterior densities which makes them suitable for nonlinear estimation problems. However, they are computationally expensive and have a larger communication overhead compared to the Kalman filtering methods. In the following sections, we present several distributed particle filtering schemes.

An article by Coates [5] provides seminal work on the distributed particle filters. The scheme presented in this paper is based on factorization of the global likelihood function. The products of factors of global likelihood function are parametrized. The particles and their associated likelihood functions are used to learn the parameters which are then exchanged between sensor nodes to compute an approximation of the global likelihood function. This scheme has a much lower communication overhead compared to any scheme that directly transmits the raw measurements when the measurement data dimensionality is higher than the dimensionality of the parameters.

In [6], Oreshkin et al. present a scheme that is based on parametrization of the local posterior. The local posterior is parametrized by a Gaussian distribution. The parameters of the local posterior densities are computed from particles and their associated likelihood functions. These parameters are then exchanged between sensor nodes using gossip algorithm and an approximation of the global posterior is computed. The gossip step is based on an optimal fusion strategy that results in lowering the communication overhead of this scheme.

In [7], Hlinka et al. propose a scheme that is based on distributed computation of the global likelihood function. This approach identifies a set of sufficient statistics that are exchanged between the sensor nodes using gossip algorithm and an approximation to the global likelihood function is computed locally. Then each sensor node uses this approximate global likelihood function to update the particle weights.

Farahmand et al. [8] propose a scheme that computes a set of samples at each sensor

node such that the set captures most of the mass in the local posterior density. Sensor nodes then employ gossip over the local sets and compute a global set at each senor node. The global sample set is constructed such that it captures most of the mass in the global posterior density. This approach has a significantly lower communication overhead compared to any scheme that gossips on all the particle weights.

In [9], Ustebay et al. present a scheme in which each sensor node computes local particle weights based on its likelihood function and local measurement. Then sensor nodes gossip on the local weights to compute the global particle weights which approximate the global posterior distribution. This approach uses a selective gossip procedure that only gossips on a subset of the local particle weights. The set of local particle weights for gossip is selected such that it leads to a set of global particle weights that captures most of the mass in the global posterior distribution. Hence, this approach also has a good communication efficiency compared to the schemes that gossip on all the particle weights.

In schemes [81, 82] the likelihood for the bearings-only model is parametrized exactly by a few sufficient statistics. These statistics are computed at all the nodes using the gossip algorithm and the particle weights are updated locally. These approaches have good communication efficiency, but are applicable only to the bearings-only measurement model.

2.4.3 Discussion

The distributed Kalman filtering schemes provide the optimal solution to the linear estimation problem and also have a low computational and communication overhead. However, these methods do not work well for non-linear and non-Gaussian problems. The distributed unscented Kalman filtering approaches provide an extension to the non-linear problems. These methods also use mean and covariance based parametrization of the posterior density, hence they also have a low computational and communication overhead. But, they have a limited ability to handle the non-linearities in the state dynamics and the measurement models. Distributed particle filtering methods have the ability to approximate more general posterior distributions and provide better estimation accuracy for non-linear problems as compared to the Kalman filtering methods. However, they have a much higher computational and communication overhead compared to the Kalman filtering methods. The distributed particle filtering schemes [5, 6] use parametric approximations of the local posterior (particle based) densities and employ gossip on these parameters. In [7], the local

likelihoods are parametrized and gossip is used to compute the parameters of the global likelihood. In both these cases, parametrization is used to reduce the communication overhead, but the estimation accuracy also decreases. The schemes [8, 83] do not use any parametrization and strive to achieve a better performance by gossiping on the particle weights, but at the cost of a higher communication overhead.

2.5 Summary

In this chapter, we provided background information about the ensemble Kalman filter (EnKF). The EnKF provides an extension of the Kalman filtering framework to problems with non-linear state dynamics. The EnKF uses a set of samples to approximate the first two moments of the posterior distribution. The samples are propagated using the state dynamics; this allows the EnKF to capture the non-linearities in the state dynamics. The EnKF requires a relatively small sample size to provide a good approximation to the mean and covariance of the posterior density. Several modifications and extensions of the EnKF have been proposed that improve its performance in specific application scenarios.

We also reviewed the distributed filtering schemes along with the decentralized computation methods employed by these schemes. The distributed Kalman filtering schemes have good computational and communication efficiency, but have a limited ability to handle non-linearities. On the other hand, the particle filtering methods work well for non-linear problems, but have a higher computational and communication overhead. Several particle filtering schemes use parametrization to reduce their communication overhead, but it also results in a lower estimation accuracy.

Chapter 3

Distributed Ensemble Kalman Filtering in Wireless Sensor Networks

In this chapter, we formulate the distributed implementations of three forms of the ensemble Kalman filter. Specifically, we propose novel distributed approximations of the ensemble Kalman filter (EnKF) [16], ensemble square root filter (ESRF) [20] and deterministic ensemble Kalman filter (DEnKF) [21]. In the proposed approach, every sensor node runs a local copy of the filter and the nodes communicate via the randomized gossip scheme [11]. We have used the randomized gossip algorithm because most of the existing distributed filtering schemes have reported results based on this algorithm. The difficulty in developing a distributed scheme lies in the fact that the analysis equations require the complete knowledge of the measurements and the measurement model parameters while the sensor nodes only possess local information. We address this challenge by expressing the analysis equations in an alternative form and identifying and gossiping on a set of statistics that are required by the sensor nodes to perform the analysis locally. We formulate the distributed filters for both the linear and the non-linear measurement models.

3.1 Problem Statement

The goal is to sequentially estimate the system state that evolves according to a discrete-time Markov process. The evolution of the state is characterized by the *system model*

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}) + \mathbf{n}_{k-1}, \tag{3.1}$$

where $\mathbf{x} \in \mathcal{R}^n$ is the system state, $f : \mathcal{R}^n \rightarrow \mathcal{R}^n$ is a nonlinear system function and $\mathbf{n}_{k-1} \in \mathcal{R}^n$ is the system noise which follows a Gaussian distribution with zero mean and covariance $\mathbf{Q}$.

A set of wireless sensor nodes V are deployed in a physical region and record measurements related to the system state. The relationship between the local measurement of a sensor node $v \in V$ and the state is specified by its *measurement model*

$$\mathbf{y}^v = h^v(\mathbf{x}) + \epsilon^v, \quad v \in V, \tag{3.2}$$

where $\mathbf{y}^v \in \mathcal{R}^p$, $h^v : \mathcal{R}^n \rightarrow \mathcal{R}^p$ and $\epsilon_k^v \in \mathcal{R}^p$ are the sensor-dependent measurement, measurement function (possibly non-linear) and measurement noise, respectively. The measurement noise for each sensor node $v \in V$ follows a Gaussian distribution with mean zero and covariance $\mathbf{R}^v$ and the sensor nodes have uncorrelated measurement noise.

We assume that the sensor nodes do not have the complete topological information and are only aware of their neighboring nodes. The nodes that have the ability to directly communicate with each other are considered as neighbors. We also assume that the sensor nodes are unaware of the measurement model parameters of the other nodes. In a special case of high dimensional measurements, we relax this condition and allow each sensor node to know all the measurement models. Consequently, the sensor nodes cannot exchange raw measurements; in the former case due to the unawareness of the measurement models, and in the latter case due to the high communication cost associated with it.

3.2 Distributed Forecast

In the proposed distributed scheme, every sensor node runs a local copy of the filter. Since, all three forms of the ensemble Kalman filter have the same forecast step, we present a common distributed forecast step for all of them. The distributed implementations of the forecast step require the random number generators of all the nodes to be synchronized (initialized with the same seed) which can be accomplished using the decentralized routine described in [84].

At the start, every node generates the initial samples at time $k = 0$ by sampling from the same prior density $p(\mathbf{x}_0)$. At each time step k, every node $v \in V$ calculates its local forecast samples $\mathbf{x}^f_{(i)k}$ by evolving its local previous time updated samples $\mathbf{x}^a_{(i)k-1}$ through the system

model function and adding a system noise realization. This forecast is given by the equation

$$\mathbf{x}^f_{(i)k} = f(\mathbf{x}^a_{(i)k-1}) + \mathbf{n}_{(i)k-1}, \quad v \in V, \tag{3.3}$$

where $f(\cdot)$ is the system function, $\mathbf{x}^a_{(i)k-1}$ is a previous time local updated sample and $\mathbf{n}_{(i)k-1}$ is a system noise realization. The system noise is sampled from a Gaussian density with mean zero and covariance $\mathbf{Q}$.

All the sensor nodes use pseudo random number generators initialized with the same seed, hence every node produces identical samples from the prior density and the noise density. This distributed forecast mechanism ensures that all the nodes will have the same local forecast samples if they had identical previous time local updated samples.

3.3 Distributed Analysis for Linear Measurement Models

The centralized measurements and measurement model parameters can be expressed in terms of the individual measurements and measurement model parameters of the sensor nodes as follows

$$\mathbf{y} = [\mathbf{y}^1; \mathbf{y}^2; \ldots; \mathbf{y}^{|V|}], \tag{3.4}$$

$$\epsilon = [\epsilon^1; \epsilon^2; \ldots; \epsilon^{|V|}], \tag{3.5}$$

$$\mathbf{H} = [\mathbf{H}^1; \mathbf{H}^2; \ldots; \mathbf{H}^{|V|}], \tag{3.6}$$

$$\mathbf{R} = \mathrm{diag}(\mathbf{R}^1, \mathbf{R}^2, \ldots, \mathbf{R}^{|V|}), \tag{3.7}$$

where $\mathbf{y}^v$ and ϵ^v are p-dimensional vectors representing the measurement and measurement noise of sensor v, $\mathbf{H}^v$ is a $p \times n$ matrix representing the linear measurement model function of sensor v and $\mathbf{R}^v$ is a $p \times p$ measurement noise covariance matrix of sensors v.

In this distributed approach, we assume that every sensor node is aware of the network size, i.e. the total number of sensor nodes in the network. The sensor nodes can use the decentralized scheme outlined in [6] to learn the network size. In the following sections, we present the distributed analysis approach for the three forms of the ensemble Kalman filter.

3.3.1 Ensemble Kalman Filter (EnKF)

In order to formulate the distributed analysis step in the ensemble Kalman filter (EnKF) we extend the information Kalman filtering [85, 86] framework to the EnKF. In the centralized problem, the EnKF analysis equations can be expressed in the information form as follows

$$\mathbf{x}^a_{(i)} = \mathbf{x}^f_{(i)} + \mathbf{K}(\mathbf{y}_{(i)} - \mathbf{H}\mathbf{x}^f_{(i)}), \tag{3.8}$$

where the Kalman gain $\mathbf{K}$, is given by

$$\mathbf{P}^a = [(\mathbf{P}^f)^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}]^{-1}, \tag{3.9}$$

$$\mathbf{K} = \mathbf{P}^a\mathbf{H}^T\mathbf{R}^{-1}. \tag{3.10}$$

We can express the above analysis equations in terms of the measurements and the measurement model parameters of the individual sensors as follows

$$\mathbf{x}^a_{(i)} = \mathbf{x}^f_{(i)} + \mathbf{K}(\mathbf{y}_{(i)} - \mathbf{H}\mathbf{x}^f_{(i)}), \tag{3.11}$$

$$= \mathbf{x}^f_{(i)} + \mathbf{P}^a\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{y}_{(i)} - \mathbf{H}\mathbf{x}^f_{(i)}), \tag{3.12}$$

$$= \mathbf{x}^f_{(i)} + \mathbf{P}^a(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{y}_{(i)} - \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{x}^f_{(i)}), \tag{3.13}$$

$$= \mathbf{x}^f_{(i)} + \mathbf{P}^a\left(\sum_{v\in V}(\mathbf{H}^v)^T(\mathbf{R}^v)^{-1}\mathbf{y}^v_{(i)} - \sum_{v\in V}(\mathbf{H}^v)^T(\mathbf{R}^v)^{-1}\mathbf{H}^v\mathbf{x}^f_{(i)}\right), \tag{3.14}$$

where $\mathbf{P}^a$ is given by

$$\mathbf{P}^a = [(\mathbf{P}^f)^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}]^{-1}, \tag{3.15}$$

$$= \left[(\mathbf{P}^f)^{-1} + \sum_{v\in V}(\mathbf{H}^v)^T(\mathbf{R}^v)^{-1}\mathbf{H}^v\right]^{-1}. \tag{3.16}$$

The distributed analysis step is devised based on the analysis equations 3.14 and 3.16. At the analysis time each sensor nodes $v \in V$ initializes its local variables as follows

$$\{\mathbf{Y}^v_{(i)}\}_{i=1}^N = \{|V|(\mathbf{H}^v)^T(\mathbf{R}^v)^{-1}\mathbf{y}^v_{(i)}\}_{i=1}^N, \tag{3.17}$$

$$\mathbf{S}^v = |V|(\mathbf{H}^v)^T(\mathbf{R}^v)^{-1}\mathbf{H}^v, \tag{3.18}$$

where N is the number of samples. Here, $\mathbf{Y}_{(i)}^{v}$ is an n-dimensional vector and $\mathbf{S}^{v}$ is an n-dimensional square matrix, where n is the number of state variables.

The sensor nodes can then perform a randomized gossip routine to compute the element-wise average of these variables over the entire sensor network. Since, the gossip routine runs for a finite number of iterations the local variables at all the sensor nodes might have slightly different values. We use a second gossip routine (max gossip) which is a simple variation of the randomized gossip routine. In each iteration of the max gossip routine both the communicating sensor nodes choose the element-wise maximum instead of the element-wise average for each variable. This ensures that with sufficient gossip, all the sensor nodes will have nearly identical local variables after the two gossip routines. In the limit of infinite gossip, these local variables will contain the following values

$$\{\widehat{\mathbf{Y}}_{(i)}\}_{i=1}^{N} = \left\{ \sum_{v \in V} (\mathbf{H}^{v})^{T} (\mathbf{R}^{v})^{-1} \mathbf{y}_{(i)}^{v} \right\}_{i=1}^{N}, \tag{3.19}$$

$$\widehat{\mathbf{S}} = \sum_{v \in V} (\mathbf{H}^{v})^{T} (\mathbf{R}^{v})^{-1} \mathbf{H}^{v}, \tag{3.20}$$

where $\widehat{\mathbf{Y}}_{(i)}$ and $\widehat{\mathbf{S}}$ are the values obtained after running the average and max gossip routines on the $\{\mathbf{Y}_{(i)}\}_{v \in V}$ and $\{\mathbf{S}\}_{v \in V}$, respectively.
After this, every sensor node can perform the analysis locally to compute identical updated samples using the equation

$$\mathbf{x}_{(i)}^{a} = \mathbf{x}_{(i)}^{f} + [(\mathbf{P}^{f})^{-1} + \widehat{\mathbf{S}}]^{-1} (\widehat{\mathbf{Y}}_{(i)} - \widehat{\mathbf{S}} \mathbf{x}_{(i)}^{f}). \tag{3.21}$$

The complete distributed EnKF filtering approach is presented in Algorithm 4.

3.3.2 Ensemble Square Root Filter (ESRF)

In a similar way, the analysis equations in the ESRF can also be written in the information form and expressed in terms of the local measurements and the measurement model

Algorithm 4 Distributed Ensemble Kalman Filter

// Initialization at time $k = 0$
for each node $v \in V$ **do**
  for each sample $i = 1, 2, .., N$ **do**
    Sample: $\mathbf{x}_{(i)0}^{a,v} \sim p(\mathbf{x}_0)$
  end for
end for

// For time $k > 0$
for $k = 1, 2, .., T$ **do**

  // Local forecast
  for each node $v \in V$ **do**
    for each sample $i = 1, 2, .., N$ **do**
      Sample: $\mathbf{n}_{(i)k-1}^{v} \sim \mathcal{N}(0, \mathbf{Q})$
      $\mathbf{x}_{(i)k}^{f,v} = f(\mathbf{x}_{(i)k-1}^{a,v}) + \mathbf{n}_{(i)k-1}^{v}$
    end for
    $\hat{\mathbf{x}}_k^{f,v} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_{(i)k}^{f,v}$
    $\mathbf{P}_k^{f,v} = \frac{1}{N-1} \sum_{i=1}^{N} (\mathbf{x}_{(i)k}^{f,v} - \hat{\mathbf{x}}_k^{f,v})(\mathbf{x}_{(i)k}^{f,v} - \hat{\mathbf{x}}_k^{f,v})^T$
  end for

  // Local variable initialization
  for each node $v \in V$ **do**
    for each sample $i = 1, 2, .., N$ **do**
      Sample: $\epsilon_{(i)k}^{v} \sim \mathcal{N}(0, \mathbf{R}_k^v)$
      $\mathbf{y}_{(i)k}^{v} = \mathbf{y}_k^v + \epsilon_{(i)k}^{v}$
      $\mathbf{Y}_{(i)k}^{v} = |V|(\mathbf{H}^{vT}(\mathbf{R}_k^v)^{-1}\mathbf{y}_{(i)k}^{v})$
    end for
    $\mathbf{S}_k^v = |V|(\mathbf{H}^{vT}(\mathbf{R}_k^v)^{-1}\mathbf{H}^v)$
  end for

  // Randomized gossip
  $[\{\widetilde{\mathbf{Y}}_{(i)k}^{v}\}_{i=1,v\in V}^{N}, \{\widetilde{\mathbf{S}}_k^v\}_{v\in V}] = \text{Average gossip}(\{\mathbf{Y}_{(i)k}^{v}\}_{i=1,v\in V}^{N}, \{\mathbf{S}_k^v\}_{v\in V})$
  $[\{\widehat{\mathbf{Y}}_{(i)k}^{v}\}_{i=1,v\in V}^{N}, \{\widehat{\mathbf{S}}_k^v\}_{v\in V}] = \text{Max gossip}(\{\widetilde{\mathbf{Y}}_{(i)k}^{v}\}_{i=1,v\in V}^{N}, \{\widetilde{\mathbf{S}}_k^v\}_{v\in V})$

  // Local analysis
  for each node $v \in V$ **do**
    $\mathbf{P}_k^{a,v} = [(\mathbf{P}_k^{f,v})^{-1} + \widehat{\mathbf{S}}_k^v]^{-1}$
    for each sample $i = 1, 2, .., N$ **do**
      $\mathbf{x}_{(i)k}^{a,v} = \mathbf{x}_{(i)k}^{f,v} + \mathbf{P}_k^{a,v}(\widehat{\mathbf{Y}}_{(i)k}^{v} - \widehat{\mathbf{S}}_k^v \mathbf{x}_{(i)k}^{f,v})$
    end for
  end for

end for

parameters of the sensors. The analysis equation for the mean is given by

$$\hat{\mathbf{x}}^a = \hat{\mathbf{x}}^f + \mathbf{K}(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}^f), \tag{3.22}$$

$$= \hat{\mathbf{x}}^f + \mathbf{P}^a\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}^f), \tag{3.23}$$

$$= \hat{\mathbf{x}}^f + \mathbf{P}^a(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{y} - \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\hat{\mathbf{x}}^f), \tag{3.24}$$

$$= \hat{\mathbf{x}}^f + \mathbf{P}^a\left(\sum_{v\in V}(\mathbf{H}^v)^T(\mathbf{R}^v)^{-1}\mathbf{y}^v - \sum_{v\in V}(\mathbf{H}^v)^T(\mathbf{R}^v)^{-1}\mathbf{H}^v\hat{\mathbf{x}}^f\right), \tag{3.25}$$

where $\mathbf{P}^a$ is given by

$$\mathbf{P}^a = [(\mathbf{P}^f)^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}]^{-1}, \tag{3.26}$$

$$= \left[(\mathbf{P}^f)^{-1} + \sum_{v\in V}(\mathbf{H}^v)^T(\mathbf{R}^v)^{-1}\mathbf{H}^v\right]^{-1}. \tag{3.27}$$

The transformation matrix in the ESRF requires the evaluation of the product of the Kalman gain $\mathbf{K}$ and the measurement model function $\mathbf{H}$. This product can be expressed as follows

$$\mathbf{KH} = (\mathbf{P}^a\mathbf{H}^T\mathbf{R}^{-1})\mathbf{H} \tag{3.28}$$

$$= \mathbf{P}^a\left(\sum_{v\in V}(\mathbf{H}^v)^T(\mathbf{R}^v)^{-1}\mathbf{H}^v\right). \tag{3.29}$$

Hence, at the analysis time each sensor node $v \in V$ initializes its local variables as follows

$$\mathbf{Y}^v = |V|(\mathbf{H}^v)^T(\mathbf{R}^v)^{-1}\mathbf{y}^v, \tag{3.30}$$

$$\mathbf{S}^v = |V|(\mathbf{H}^v)^T(\mathbf{R}^v)^{-1}\mathbf{H}^v. \tag{3.31}$$

The sensor nodes perform two randomized gossip routines (average and max gossip) in the similar manner as in the distributed EnKF. After the gossip routines, the local variables at all the sensor nodes will have the values

$$\widehat{\mathbf{Y}} = \sum_{v\in V}(\mathbf{H}^v)^T(\mathbf{R}^v)^{-1}\mathbf{y}^v, \tag{3.32}$$

$$\widehat{\mathbf{S}} = \sum_{v\in V}(\mathbf{H}^v)^T(\mathbf{R}^v)^{-1}\mathbf{H}^v. \tag{3.33}$$

The mean of the updated samples can then be calculated at each sensor node using the following equation

$$\hat{\mathbf{x}}^a = \hat{\mathbf{x}}^f + [(\mathbf{P}^f)^{-1} + \widehat{\mathbf{S}}]^{-1}(\widehat{\mathbf{Y}} - \widehat{\mathbf{S}}\hat{\mathbf{x}}^f). \tag{3.34}$$

Each sensor node can then compute the transformation matrix $\mathbf{T}$ as follows

$$\mathbf{T} = \left(\mathbf{I} - [(\mathbf{P}^f)^{-1} + \widehat{\mathbf{S}}]^{-1}\widehat{\mathbf{S}}\right)^{1/2}. \tag{3.35}$$

After the computation of the updated mean and the transformation matrix each sensor node updates the samples using the equation

$$\mathbf{x}^a_{(i)} = \hat{\mathbf{x}}^a + \mathbf{T}(\mathbf{x}^f_{(i)} - \hat{\mathbf{x}}^f). \tag{3.36}$$

This distributed filtering methodology is summarized in the Algorithm 5.

3.3.3 Deterministic Ensemble Kalman Filter (DEnKF)

In the distributed DEnKF analysis scheme, exactly the same steps as the ESRF are followed for the local forecast, local variable initialization and gossip procedures. The analysis scheme in the EnKF differs only in the way the transformation matrix is computed. In this scheme, the transformation matrix can be computed at each sensor node using its local variables as follows

$$\mathbf{T} = \mathbf{I} - \frac{1}{2}[(\mathbf{P}^f)^{-1} + \widehat{\mathbf{S}}]^{-1}\widehat{\mathbf{S}}. \tag{3.37}$$

The distributed DEnKF filtering approach is presented in Algorithm 6.

3.4 Distributed Analysis for Non-linear Measurement Models

The analysis scheme in the ensemble Kalman filters (EnKFs) requires the measurement model to be linear. We explore three strategies for incorporating non-linear measurement models in the EnKF analysis. The linearization approaches, namely linearization at sample mean, pseudo measurement matrix representation and linearization at each sample are explained in Appendix A.1. In the subsequent sections, we formulate the distributed analysis

Algorithm 5 Distributed Ensemble Square Root Filter

// Initialization at time $k = 0$
for each node $v \in V$ **do**
  for each sample $i = 1, 2, .., N$ **do**
    Sample: $\mathbf{x}_{(i)0}^{a,v} \sim p(\mathbf{x}_0)$
  end for
end for

// For time $k > 0$
for $k = 1, 2, .., T$ **do**

  // Local forecast
  for each node $v \in V$ **do**
    for each sample $i = 1, 2, .., N$ **do**
      Sample: $\mathbf{n}_{(i)k-1}^{v} \sim \mathcal{N}(0, \mathbf{Q})$
      $\mathbf{x}_{(i)k}^{f,v} = f(\mathbf{x}_{(i)k-1}^{a,v}) + \mathbf{n}_{(i)k-1}^{v}$
    end for
    $\hat{\mathbf{x}}_k^{f,v} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_{(i)k}^{f,v}$
    $\mathbf{P}_k^{f,v} = \frac{1}{N-1} \sum_{i=1}^{N} (\mathbf{x}_{(i)k}^{f,v} - \hat{\mathbf{x}}_k^{f,v})(\mathbf{x}_{(i)k}^{f,v} - \hat{\mathbf{x}}_k^{f,v})^T$
  end for

  // Local variable initialization
  for each node $v \in V$ **do**
    $\mathbf{Y}_k^v = |V|(\mathbf{H}^{vT}(\mathbf{R}_k^v)^{-1}\mathbf{y}_k^v)$
    $\mathbf{S}_k^v = |V|(\mathbf{H}^{vT}(\mathbf{R}_k^v)^{-1}\mathbf{H}^v)$
  end for

  // Randomized gossip
  $[\{\widetilde{\mathbf{Y}}_k^v\}_{v\in V}, \{\widetilde{\mathbf{S}}_k^v\}_{v\in V}] = \text{Average gossip}(\{\mathbf{Y}_k^v\}_{v\in V}, \{\mathbf{S}_k^v\}_{v\in V})$
  $[\{\widehat{\mathbf{Y}}_k^v\}_{v\in V}, \{\widehat{\mathbf{S}}_k^v\}_{v\in V}] = \text{Max gossip}(\{\widetilde{\mathbf{Y}}_k^v\}_{v\in V}, \{\widetilde{\mathbf{S}}_k^v\}_{v\in V})$

  // Local analysis
  for each node $v \in V$ **do**
    $\mathbf{P}_k^{a,v} = [(\mathbf{P}_k^{f,v})^{-1} + \widehat{\mathbf{S}}_k^v]^{-1}$
    $\mathbf{T}_k^v = (\mathbf{I} - \mathbf{P}_k^{a,v}\widehat{\mathbf{S}}_k^v)^{1/2}$
    $\hat{\mathbf{x}}_k^{a,v} = \hat{\mathbf{x}}_k^{f,v} + \mathbf{P}_k^{a,v}(\widehat{\mathbf{Y}}_k^v - \widehat{\mathbf{S}}_k^v\hat{\mathbf{x}}_k^{f,v})$
    for each sample $i = 1, 2, .., N$ **do**
      $\mathbf{x}_{(i)k}^{a,v} = \hat{\mathbf{x}}_k^{a,v} + \mathbf{T}_k^v(\mathbf{x}_{(i)k}^{f,v} - \hat{\mathbf{x}}_k^{f,v})$
    end for
  end for

end for

Algorithm 6 Distributed Deterministic Ensemble Kalman Filter

// Initialization at time $k = 0$
for each node $v \in V$ **do**
  for each sample $i = 1, 2, .., N$ **do**
    Sample: $\mathbf{x}_{(i)0}^{a,v} \sim p(\mathbf{x}_0)$
  end for
end for

// For time $k > 0$
for $k = 1, 2, .., T$ **do**

  // Local forecast
  for each node $v \in V$ **do**
    for each sample $i = 1, 2, .., N$ **do**
      Sample: $\mathbf{n}_{(i)k-1}^{v} \sim \mathcal{N}(0, \mathbf{Q})$
      $\mathbf{x}_{(i)k}^{f,v} = f(\mathbf{x}_{(i)k-1}^{a,v}) + \mathbf{n}_{(i)k-1}^{v}$
    end for
    $\hat{\mathbf{x}}_k^{f,v} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_{(i)k}^{f,v}$
    $\mathbf{P}_k^{f,v} = \frac{1}{N-1} \sum_{i=1}^{N} (\mathbf{x}_{(i)k}^{f,v} - \hat{\mathbf{x}}_k^{f,v})(\mathbf{x}_{(i)k}^{f,v} - \hat{\mathbf{x}}_k^{f,v})^T$
  end for

  // Local variable initialization
  for each node $v \in V$ **do**
    $\mathbf{Y}_k^v = |V|(\mathbf{H}^{vT}(\mathbf{R}_k^v)^{-1}\mathbf{y}_k^v)$
    $\mathbf{S}_k^v = |V|(\mathbf{H}^{vT}(\mathbf{R}_k^v)^{-1}\mathbf{H}^v)$
  end for

  // Randomized gossip
  $[\{\widetilde{\mathbf{Y}}_k^v\}_{v\in V}, \{\widetilde{\mathbf{S}}_k^v\}_{v\in V}] = \text{Average gossip}(\{\mathbf{Y}_k^v\}_{v\in V}, \{\mathbf{S}_k^v\}_{v\in V})$
  $[\{\widehat{\mathbf{Y}}_k^v\}_{v\in V}, \{\widehat{\mathbf{S}}_k^v\}_{v\in V}] = \text{Max gossip}(\{\widetilde{\mathbf{Y}}_k^v\}_{v\in V}, \{\widetilde{\mathbf{S}}_k^v\}_{v\in V})$

  // Local analysis
  for each node $v \in V$ **do**
    $\mathbf{P}_k^{a,v} = [(\mathbf{P}_k^{f,v})^{-1} + \widehat{\mathbf{S}}_k^v]^{-1}$
    $\mathbf{T}_k^v = \mathbf{I} - \frac{1}{2}\mathbf{P}_k^{a,v}\widehat{\mathbf{S}}_k^v$
    $\hat{\mathbf{x}}_k^{a,v} = \hat{\mathbf{x}}_k^{f,v} + \mathbf{P}_k^{a,v}(\widehat{\mathbf{Y}}_k^v - \widehat{\mathbf{S}}_k^v\hat{\mathbf{x}}_k^{f,v})$
    for each sample $i = 1, 2, .., N$ **do**
      $\mathbf{x}_{(i)k}^{a,v} = \hat{\mathbf{x}}_k^{a,v} + \mathbf{T}_k^v(\mathbf{x}_{(i)k}^{f,v} - \hat{\mathbf{x}}_k^{f,v})$
    end for
  end for

end for

schemes for the three forms of the EnKF using each of these linearization strategies. Two of these linearization strategies (linearization at the sample mean and the pseudo measurement matrix representation) compute a single linearized approximation that is common to all the samples. The approximation can differ, but thereafter the distributed analysis scheme is the same for the two strategies, so we group them together below.

3.4.1 Ensemble Kalman Filter (EnKF)

In case of the linearization at the sample mean and pseudo measurement matrix representation all of the samples use a common linearized model. In this scenario, we can write the EnKF analysis equations in the information form and express them in terms of the individual measurement models of the sensor nodes as follows

$$\mathbf{P}^a = \left[(\mathbf{P}^f)^{-1} + \sum_{v \in V} (\mathbf{H}^v)^T (\mathbf{R}^v)^{-1} \mathbf{H}^v \right]^{-1}, \tag{3.38}$$

$$\mathbf{x}^a_{(i)} = \mathbf{x}^f_{(i)} + \mathbf{P}^a \left(\sum_{v \in V} (\mathbf{H}^v)^T (\mathbf{R}^v)^{-1} \mathbf{y}^v_{(i)} - \sum_{v = \in V} (\mathbf{H}^v)^T (\mathbf{R}^v)^{-1} h^v(\mathbf{x}^f_{(i)}) \right), \tag{3.39}$$

where $\mathbf{H}^v$ is the linearized measurement model of the sensor v. Here, it is assumed that a linearized approximation of each sensor's measurement model is available.
Hence, the distributed analysis scheme requires the sensor nodes to initialize the following local variables

$$\left\{ \mathbf{Y}^v_{(i)} \right\}_{i=1}^N = \left\{ |V| \left[(\mathbf{H}^v)^T (\mathbf{R}^v)^{-1} \mathbf{y}^v_{(i)} - (\mathbf{H}^v)^T (\mathbf{R}^v)^{-1} h^v(\mathbf{x}^f_{(i)}) \right] \right\}_{i=1}^N, \tag{3.40}$$

$$\mathbf{S}^v = |V| (\mathbf{H}^v)^T (\mathbf{R}^v)^{-1} \mathbf{H}^v. \tag{3.41}$$

The sensor nodes can then run average and max gossip routines on these local variables to compute the average over the network. Afterwards, every sensor node can perform the local analysis using the equation

$$\mathbf{x}^a_{(i)} = \mathbf{x}^f_{(i)} + [(\mathbf{P}^f)^{-1} + \widehat{\mathbf{S}}]^{-1} (\widehat{\mathbf{Y}}_{(i)}). \tag{3.42}$$

This distributed analysis procedure is presented in Algorithm 7.

A similar procedure can be used to develop distributed analysis step for the case where

Algorithm 7 Distributed EnKF Analysis (Common linearization)

$$\left[\{\mathbf{x}_{(i)}^{a,v}\}_{i=1,v\in V}^{N}\right] = \text{EnKF Analysis}\left(\{\mathbf{x}_{(i)}^{f,v}\}_{i=1,v\in V}^{N}, \{\mathbf{P}^{f,v}\}_{v\in V}, \{\mathbf{H}^{v}\}_{v\in V}\right)$$

1: `// Local variable initialization`
2: **for** each node $v \in V$ **do**
3: **for** each sample $i = 1, 2, .., N$ **do**
4: Sample: $\epsilon_{(i)}^{v} \sim \mathcal{N}(0, \mathbf{R}_k^v)$
5: $\mathbf{y}_{(i)}^{v} = \mathbf{y}^{v} + \epsilon_{(i)}^{v}$
6: $\mathbf{Y}_{(i)}^{v} = |V|(\mathbf{H}^{vT}(\mathbf{R}_k^v)^{-1}\mathbf{y}_{(i)}^{v} - \mathbf{H}^{vT}(\mathbf{R}_k^v)^{-1}h^v(\mathbf{x}_{(i)}^{f,v}))$
7: **end for**
8: $\mathbf{S}^v = |V|(\mathbf{H}^{vT}(\mathbf{R}_k^v)^{-1}\mathbf{H}^v)$
9: **end for**

10: `// Randomized gossip`
11: $[\{\widetilde{\mathbf{Y}}_{(i)}^{v}\}_{i=1,v\in V}^{N}, \{\widetilde{\mathbf{S}}^{v}\}_{v\in V}] = \text{Average gossip}(\{\mathbf{Y}_{(i)}^{v}\}_{i=1,v\in V}^{N}, \{\mathbf{S}^{v}\}_{v\in V})$
12: $[\{\widehat{\mathbf{Y}}_{(i)}^{v}\}_{i=1,v\in V}^{N}, \{\widehat{\mathbf{S}}^{v}\}_{v\in V}] = \text{Max gossip}(\{\widetilde{\mathbf{Y}}_{(i)}^{v}\}_{i=1,v\in V}^{N}, \{\widetilde{\mathbf{S}}^{v}\}_{v\in V})$

13: `// Local analysis`
14: **for** each node $v \in V$ **do**
15: $\mathbf{P}^{a,v} = [(\mathbf{P}^{f,v})^{-1} + \widehat{\mathbf{S}}^{v}]^{-1}$
16: **for** each sample $i = 1, 2, .., N$ **do**
17: $\mathbf{x}_{(i)}^{a,v} = \mathbf{x}_{(i)}^{f,v} + \mathbf{P}^{a,v}(\widehat{\mathbf{Y}}_{(i)}^{v})$
18: **end for**
19: **end for**

the measurement model is linearized at each sample point. In this situation, the analysis equation has the form

$$\mathbf{P}_{(i)}^{a} = \left[(\mathbf{P}^{f})^{-1} + \sum_{v\in V}(\mathbf{H}_{(i)}^{v})^{T}(\mathbf{R}^{v})^{-1}\mathbf{H}_{(i)}^{v}\right]^{-1}, \tag{3.43}$$

$$\mathbf{x}_{(i)}^{a} = \mathbf{x}_{(i)}^{f} + \mathbf{P}_{(i)}^{a}\left(\sum_{v\in V}(\mathbf{H}_{(i)}^{v})^{T}(\mathbf{R}^{v})^{-1}\mathbf{y}_{(i)}^{v} - \sum_{v=\in V}(\mathbf{H}_{(i)}^{v})^{T}(\mathbf{R}^{v})^{-1}h^{v}(\mathbf{x}_{(i)}^{f})\right), \tag{3.44}$$

where $\mathbf{H}_{(i)}^{v}$ is a linearization of sensor v's measurement model performed at the sample $\mathbf{x}_{(i)}^{f}$.

Algorithm 8 Distributed EnKF Analysis (Linearization at each sample)

$$\left[\{\mathbf{x}_{(i)}^{a,v}\}_{i=1,v\in V}^{N}\right] = \text{EnKF Analysis}\left(\{\mathbf{x}_{(i)}^{f,v}\}_{i=1,v\in V}^{N}, \{\mathbf{P}^{f,v}\}_{v\in V}, \{\mathbf{H}_{(i)}^{v}\}_{i=1,v\in V}^{N}\right)$$

1: `// Local variable initialization`
2: **for** each node $v \in V$ **do**
3: **for** each sample $i = 1, 2, .., N$ **do**
4: Sample: $\epsilon_{(i)}^{v} \sim \mathcal{N}(0, \mathbf{R}_k^v)$
5: $\mathbf{y}_{(i)}^{v} = \mathbf{y}^{v} + \epsilon_{(i)}^{v}$
6: $\mathbf{Y}_{(i)}^{v} = |V|(\mathbf{H}_{(i)}^{vT}(\mathbf{R}_k^v)^{-1}\mathbf{y}_{(i)}^{v} - \mathbf{H}_{(i)}^{vT}(\mathbf{R}_k^v)^{-1}h^{v}(\mathbf{x}_{(i)}^{f,v}))$
7: $\mathbf{S}_{(i)}^{v} = |V|(\mathbf{H}_{(i)}^{vT}(\mathbf{R}_k^v)^{-1}\mathbf{H}_{(i)}^{v})$
8: **end for**
9: **end for**
10: `// Randomized gossip`
11: $[\{\widetilde{\mathbf{Y}}_{(i)}^{v}\}_{i=1,v\in V}^{N}, \{\widetilde{\mathbf{S}}_{(i)}^{v}\}_{i=1,v\in V}^{N}] = \text{Average gossip}(\{\mathbf{Y}_{(i)}^{v}\}_{i=1,v\in V}^{N}, \{\mathbf{S}_{(i)}^{v}\}_{i=1,v\in V}^{N})$
12: $[\{\widehat{\mathbf{Y}}_{(i)}^{v}\}_{i=1,v\in V}^{N}, \{\widehat{\mathbf{S}}_{(i)}^{v}\}_{i=1,v\in V}^{N}] = \text{Max gossip}(\{\widetilde{\mathbf{Y}}_{(i)}^{v}\}_{i=1,v\in V}^{N}, \{\widetilde{\mathbf{S}}_{(i)}^{v}\}_{i=1,v\in V}^{N})$
13: `// Local analysis`
14: **for** each node $v \in V$ **do**
15: **for** each sample $i = 1, 2, .., N$ **do**
16: $\mathbf{P}_{(i)}^{a,v} = [(\mathbf{P}^{f,v})^{-1} + \widehat{\mathbf{S}}_{(i)}^{v}]^{-1}$
17: $\mathbf{x}_{(i)}^{a,v} = \mathbf{x}_{(i)}^{f,v} + \mathbf{P}_{(i)}^{a,v}(\widehat{\mathbf{Y}}_{(i)}^{v})$
18: **end for**
19: **end for**

In this case, the sensor nodes initialize the following local variables

$$\left\{\mathbf{Y}_{(i)}^{v}\right\}_{i=1}^{N} = \left\{|V|\left[(\mathbf{H}_{(i)}^{v})^{T}(\mathbf{R}^{v})^{-1}\mathbf{y}_{(i)}^{v} - (\mathbf{H}^{v})^{T}(\mathbf{R}^{v})^{-1}h_{(i)}^{v}(\mathbf{x}_{(i)}^{f})\right]\right\}_{i=1}^{N}, \tag{3.45}$$

$$\left\{\mathbf{S}_{(i)}^{v}\right\}_{i=1}^{N} = \left\{|V|(\mathbf{H}_{(i)}^{v})^{T}(\mathbf{R}^{v})^{-1}\mathbf{H}_{(i)}^{v}\right\}_{i=1}^{N}. \tag{3.46}$$

After gossiping on these variables all the sensor nodes perform the following analysis

$$\mathbf{x}_{(i)}^{a} = \mathbf{x}_{(i)}^{f} + [(\mathbf{P}^{f})^{-1} + \widehat{\mathbf{S}}_{(i)}]^{-1}(\widehat{\mathbf{Y}}_{(i)}), \tag{3.47}$$

This distributed analysis approach is summarized in Algorithm 8.

3.4.2 Ensemble Square Root Filter (ESRF)

The same linearization mechanism can be employed to develop the distributed ESRF analysis equations. In case of a common linearization, the following local variable initializations are required

$$\mathbf{Y}^v = |V|[(\mathbf{H}^v)^T(\mathbf{R}^v)^{-1}\mathbf{y}^v - (\mathbf{H}^v)^T(\mathbf{R}^v)^{-1}h^v(\hat{\mathbf{x}}^f)], \tag{3.48}$$

$$\mathbf{S}^v = |V|(\mathbf{H}^v)^T(\mathbf{R}^v)^{-1}\mathbf{H}^v. \tag{3.49}$$

The analysis step can then be implemented at every sensor node using the equations

$$\hat{\mathbf{x}}^a = \hat{\mathbf{x}}^f + [(\mathbf{P}^f)^{-1} + \widehat{\mathbf{S}}]^{-1}(\widehat{\mathbf{Y}}), \tag{3.50}$$

$$\mathbf{T} = \left(\mathbf{I} - [(\mathbf{P}^f)^{-1} + \widehat{\mathbf{S}}]^{-1}\widehat{\mathbf{S}}\right)^{1/2}, \tag{3.51}$$

$$\mathbf{x}^a_{(i)} = \hat{\mathbf{x}}^a + \mathbf{T}(\mathbf{x}^f_{(i)} - \hat{\mathbf{x}}^f), \tag{3.52}$$

where $\widehat{\mathbf{S}}$ and $\widehat{\mathbf{Y}}$ are the values obtained after running the average and max gossip routines on the $\{\mathbf{S}^v\}_{v\in V}$ and $\{\mathbf{Y}^v\}_{v\in V}$, respectively.

When linerization is performed at each sample point there is a need to compute a transformation matrix corresponding to each sample. Also, the analysis equation that computes the mean of the updated samples requires a linearized approximation of the measurement model at the mean of the forecast samples. Hence, the following variable initializations are required at every sensor node

$$\mathbf{Y}^v = |V|\left[(\mathbf{H}^v)^T(\mathbf{R}^v)^{-1}\mathbf{y}^v - (\mathbf{H}^v)^T(\mathbf{R}^v)^{-1}h^v(\hat{\mathbf{x}}^f)\right], \tag{3.53}$$

$$\mathbf{S}^v = |V|(\mathbf{H}^v)^T(\mathbf{R}^v)^{-1}\mathbf{H}^v, \tag{3.54}$$

$$\left\{\mathbf{S}^v_{(i)}\right\}_{i=1}^N = \left\{|V|(\mathbf{H}^v_{(i)})^T(\mathbf{R}^v)^{-1}\mathbf{H}^v_{(i)}\right\}_{i=1}^N, \tag{3.55}$$

where $\mathbf{H}^v$ and $\mathbf{H}^v_{(i)}$ are the linearized approximations of the sensor node v's measurement model performed at $\hat{\mathbf{x}}^f$ and $\mathbf{x}^f_{(i)}$, respectively.

Similar to the case of common linearization, the analysis can then be performed as follows

$$\hat{\mathbf{x}}^a = \hat{\mathbf{x}}^f + [(\mathbf{P}^f)^{-1} + \widehat{\mathbf{S}}]^{-1}(\widehat{\mathbf{Y}}), \tag{3.56}$$

$$\mathbf{T}_{(i)} = \left(\mathbf{I} - [(\mathbf{P}^f)^{-1} + \widehat{\mathbf{S}}_{(i)}]^{-1}\widehat{\mathbf{S}}_{(i)}\right)^{1/2}, \tag{3.57}$$

$$\mathbf{x}^a_{(i)} = \hat{\mathbf{x}}^a + \mathbf{T}_{(i)}(\mathbf{x}^f_{(i)} - \hat{\mathbf{x}}^f), \tag{3.58}$$

where $\widehat{\mathbf{S}}$, $\widehat{\mathbf{S}}_{(i)}$ and $\widehat{\mathbf{Y}}$ are the values obtained after running the average and max gossip routines on the $\{\mathbf{S}^v\}_{v\in V}$, $\{\mathbf{S}^v_{(i)}\}_{v\in V}$ and $\{\mathbf{Y}^v\}_{v\in V}$, respectively.

3.4.3 Deterministic Ensemble Kalman Filter (DEnKF)

The deterministic ensemble Kalman uses the same variable initializations and the analysis equations as the ESRF for both the common linearization approach and the individual sample based linearization approach. The only difference is in the computation of the transformation matrix. In the case of the common linearization approach, the transformation matrix is given by

$$\mathbf{T} = \mathbf{I} - \frac{1}{2}[(\mathbf{P}^f)^{-1} + \widehat{\mathbf{S}}]^{-1}\widehat{\mathbf{S}}. \tag{3.59}$$

For the sample based linearization approach the transformation matrix is evaluated as

$$\mathbf{T}_{(i)} = \mathbf{I} - \frac{1}{2}[(\mathbf{P}^f)^{-1} + \widehat{\mathbf{S}}_{(i)}]^{-1}\widehat{\mathbf{S}}_{(i)}. \tag{3.60}$$

3.5 Summary

In this chapter, we presented novel distributed approximations of three forms of the ensemble Kalman filter. The main challenge in devising a distributed approximation lies in the fact that the analysis equations require the complete knowledge of the sensor model parameters while the sensor nodes are only aware of the local model parameters. We addressed this challenge by expressing the analysis equations in an alternative from and identifying and gossiping on the sufficient statistics. We also presented three measurement model linearization strategies for the EnKFs and developed distributed approximations for each strategy.

Chapter 4

Simulations and Results

In this chapter, we evaluate the performance of the ensemble Kalman filtering schemes by applying them to a target tracking problem. We perform a set of simulations in the Matlab software (version R2013a) [87] by generating the target tracks and artificial measurements using a specific system model and several measurement models. In this study we consider three measurement models and compare the performance of the centralized and distributed filtering methods for each measurement model. In Section 4.1, the target tracking problem is explained along with a description of the system model and the measurement models used in this study. Section 4.2 provides the performance comparisons for the centralized and distributed filtering schemes.

4.1 Target Tracking Problem

We consider the problem of tracking a maneuvering target using a set of sensor nodes V. The sensor nodes are deployed in a square region and they are capable of acquiring measurements using different measurement modalities. It is assumed that the coordinates of every sensor node $v \in V$, $\mathbf{s}^v = [s_x^v, s_y^v]^T$ do not change during the course of the simulation.

4.1.1 System Model

The state of the target is defined as

$$\mathbf{x} = [x_1, x_2, x_3, x_4]^T, \tag{4.1}$$

where x_1 and x_2 represent the position of target along the x-axis and y-axis and x_3 and x_4 correspond to the velocity of target along x-axis and y-axis, respectively. The target moves in a two-dimensional square region according to the clockwise coordinated turn (CT) model [88]. The system model for the target is given by the equation

$$\mathbf{x}_k = \mathbf{A}_k \mathbf{x}_{k-1} + \begin{bmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ T & 0 \end{bmatrix} \begin{bmatrix} n_x \\ n_y \end{bmatrix}, \tag{4.2}$$

where $\mathbf{A}_k$ is the transition matrix, $\mathbf{x}_k$ and $\mathbf{x}_{k-1}$ are the state vectors of the target at time k and $k-1$, respectively, T is the sampling interval and n_x and n_y represent the zero mean Gaussian noise with variances σ_x^2 and σ_y^2, respectively. Here, n_x and n_y are assumed to be independent.

The transition matrix for the coordinated turn model is given by

$$\mathbf{A}_k = \begin{bmatrix} 1 & 0 & \frac{sin(\Omega_k T)}{\Omega_k} & -\frac{1-cos(\Omega_k T)}{\Omega_k} \\ 0 & 1 & \frac{1-cos(\Omega_k T)}{\Omega_k} & \frac{sin(\Omega_k T)}{\Omega_k} \\ 0 & 0 & cos(\Omega_k T) & -sin(\Omega_k T) \\ 0 & 0 & sin(\Omega_k T) & cos(\Omega_k T) \end{bmatrix}, \tag{4.3}$$

where $\Omega_k = c/(\sqrt{(x_{k,3})^2 + (x_{k,4})^2})$ and c is a positive constant. The clockwise coordinated turn models is a non-linear function of the system state.

4.1.2 Measurement Models

We consider three measurement modalities in this study, a) *linear* measurements, b) *range* measurements and c) *radio frequency tomography* measurements. In the case of the linear and range modalities, sensor node that has the target in its sensing range acquires a measurement. For the radio frequency (RF) tomography model all the sensors nodes acquire high-dimensional measurements. We assume that in a single simulation all the sensor nodes use the same measurement model with identical measurement noise statistics. In the subsequent sections, we present the three measurement models under consideration.

Linear measurement model

The linear model function for sensor v, $h^v(\cdot)$ is defined by a matrix $\mathbf{H}^v$. We use the following measurement matrix

$$\mathbf{H}^v = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{4.4}$$

The measurement acquired by the sensor v is given by

$$\mathbf{y}^v = \mathbf{H}^v\mathbf{x} - \mathbf{s}^v + \epsilon^v, \tag{4.5}$$

where $\mathbf{x}$ is the state vector, $\mathbf{s}^v = [s_x^v, s_y^v]^T$ is the location of sensor v and ϵ^v is the measurement noise of sensor v. The measurement noise is assumed to have a Gaussian distribution with mean zero and covariance $\mathbf{R}^v$ which is given by

$$\mathbf{R}^v = \begin{bmatrix} \sigma_\epsilon^2 & 0 \\ 0 & \sigma_\epsilon^2 \end{bmatrix}. \tag{4.6}$$

Range measurement model

The range measurement model function is defined as follows

$$h^v(\mathbf{x}) = \sqrt{(x_1 - s_x^v)^2 + (x_2 - s_y^v)^2}, \tag{4.7}$$

where x_1 and x_2 represent the position of target along the x-axis and y-axis and s_x^v and s_y^v represent the position of sensor v along the x-axis and y-axis, respectively.
The local measurement of the sensor v is given by

$$y^v = h^v(\mathbf{x}) + \epsilon^v, \tag{4.8}$$

where ϵ^v is a Gaussian noise with mean zero and variance σ_ϵ^2. The linearized approximation of this model is presented in Appendix A.2.1.

Radio Frequency (RF) Tomography measurement model

We use the radio frequency (RF) tomography model [89, 90] to demonstrate the scenario where the sensor nodes have high dimensional measurements. This model is based on the

premise that a target near the line-of-sight path between two sensor nodes will cause a signal attenuation and consequently, reduce the recorded received signal strength (RSS) value at both the nodes. In this model, each sensor node records a RSS value corresponding to every other node in the network. Hence, the dimensionality of the local measurement of a sensor node scales linearly with the the network size.

According to this model the attenuation in the signal strength between the sensor nodes u and v is given by the equation

$$A^{(u,v)}(\mathbf{x}) = \phi \exp\left(-\frac{\lambda^{(u,v)}(\mathbf{x})^2}{2\sigma_\lambda^2}\right), \tag{4.9}$$

where $\lambda^{(u,v)}(\mathbf{x})$ is a function of location of the target and the sensor nodes and ϕ and σ_λ are known parameters of the model. The function $\lambda^{(u,v)}(\mathbf{x})$ is defined as follows

$$\lambda^{(u,v)}(\mathbf{x}) = d^u(\mathbf{x}) + d^v(\mathbf{x}) - d^{(u,v)}, \tag{4.10}$$

where $d^u(\mathbf{x})$ and $d^v(\mathbf{x})$ represent the distances of the target from the sensors u and v, respectively and $d^{(u,v)}$ is the distance between the sensors.
The local measurement of the sensor node u corresponding to the node v is given by

$$y^{(u,v)} = R^{(u,v)} - A^{(u,v)}(\mathbf{x}) + \epsilon^{(u,v)}, \tag{4.11}$$

where $R^{(u,v)}$ is the reference RRS value recorded in the absence of the target and $\epsilon^{(u,v)}$ is a Gaussian measurement noise with mean zero and variance σ_ϵ^2. The measurement noise for each pair of nodes is assumed to be uncorrelated with all the other pairs. The set of measurements acquired by the sensor node v is given by $\mathbf{y}^v = [y^{(v,1)}, y^{(v,2)}, \ldots, y^{(v,|V|-1)}]^T$, where V is the set of sensor nodes. The linearized approximation of this model is presented in Appendix A.2.2.

4.2 Simulations

In order to compare the performance of the proposed filtering schemes with the existing schemes we conduct a set of numerical simulations in Matlab. In the following sections we explain the simulation setup and present the simulation results along with a discussion on

these results.

4.2.1 Target Trajectory and Sensor Deployment

We consider two simulation scenarios in this study. In the case of the linear and range measurement model we simulate a sensor network consisting of 25 sensor nodes that are deployed in a two-dimensional square region of length 50m. The sensor nodes are distributed on a grid covering this square region. In this case, every sensor node can perform sensing within a circular region around itself. The sensing radius of all the nodes is assumed to be the same and equal to 10m. Hence, only the sensor nodes that have the target in their sensing range acquire a measurement. In the case of the RF tomography measurement model, 24 sensor nodes are deployed on the perimeter of the square region such that they have uniform spacing. In this case all the sensor nodes acquire high dimensional measurements (23 scalar values) at each time step. In both the simulation scenarios, a node can only communicate with other nodes that are within its transmitting range which is set to 15m. The target moves in the square region according to the clockwise coordinated turn (CT) model described previously. The sampling time T and the model constant c are both set to 1. The target trajectory is allowed to evolve for 50 time steps using the CT model with $\sigma_x = 0.25$ and $\sigma_y = 0.25$. Figure 4.1 and Figure 4.2 show the target trajectories and the sensor deployments for the linear/range and the RF tomography measurement models, respectively.

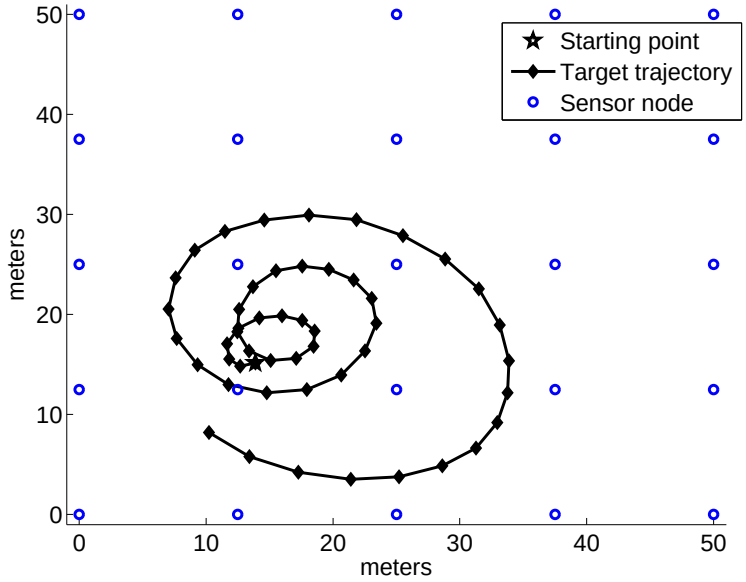

Fig. 4.1 Sensor deployment and the target trajectory for the linear and range measurement model.

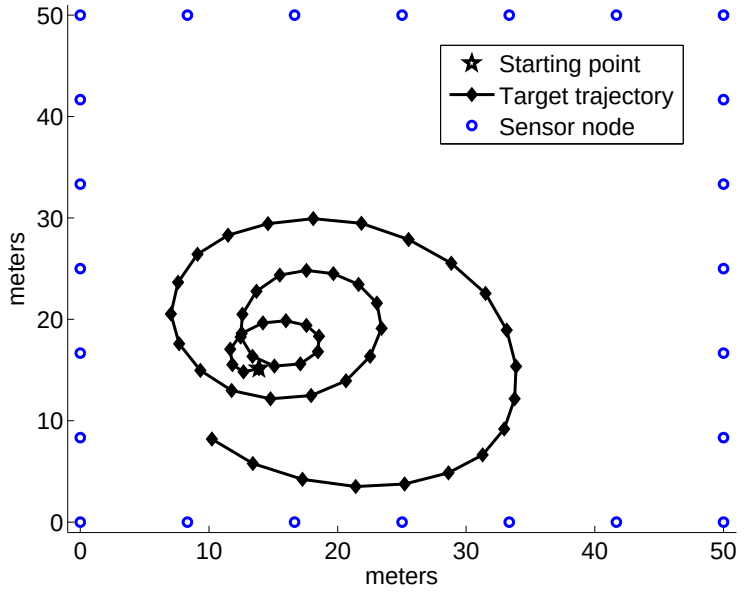

Fig. 4.2 Sensor deployment and the target trajectory for the RF tomography measurement model.

4.2.2 Simulation Settings and Error Definition

We perform a set of simulations to compare the performance of the centralized and the distributed filters. In order to have consistency, the results are averaged over 100 Monte Carlo trials. In each trial, a different measurement realization and a different sample initialization are generated. In all the simulations, the initial samples for each filter are generated by sampling from a Gaussian distribution with mean equal to the initial state of the target and standard deviation equal to 0.25 and 1 for the position and velocity state variables, respectively. The standard deviations are chosen such that a tight initialization around the true target state is obtained and hence the error at the first few time steps does not play a huge role in determining the performance of the filters. Although, the initialization has an impact on the error at the first few time steps, this does not affect the relative performance of the filters. The error in the estimated target trajectory is calculated based on the root mean square distance between the estimated target location and the true target location. The root mean square (RMS) error after k time steps is given by

$$\text{RMS error} = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(x_{i,1}-\hat{x}_{i,1})^2+(x_{i,2}-\hat{x}_{i,2})^2}, \tag{4.12}$$

where $x_{i,1}$ and $x_{i,2}$ are the true locations and $\hat{x}_{i,1}$ and $\hat{x}_{i,2}$ are the estimated locations of target along the x-axis and the y-axis, respectively at time step i. If the RMS error of a trial exceeds 2m we consider it as a lost track. The lost tracks are not included in the calculation of the average RMS error and are reported separately.

4.2.3 Performance Comparison of Centralized Filters

In the centralized setting it is assumed that all the measurements and measurement model parameters are available at a central location. In this setting, we compare the performance of the bootstrap particle filter (BPF), auxiliary particle filter (APF), unscented Kalman filter (UKF), ensemble Kalman filter (EnKF), ensemble square root filter (ESRF) and the deterministic ensemble Kalman filter (DEnKF).

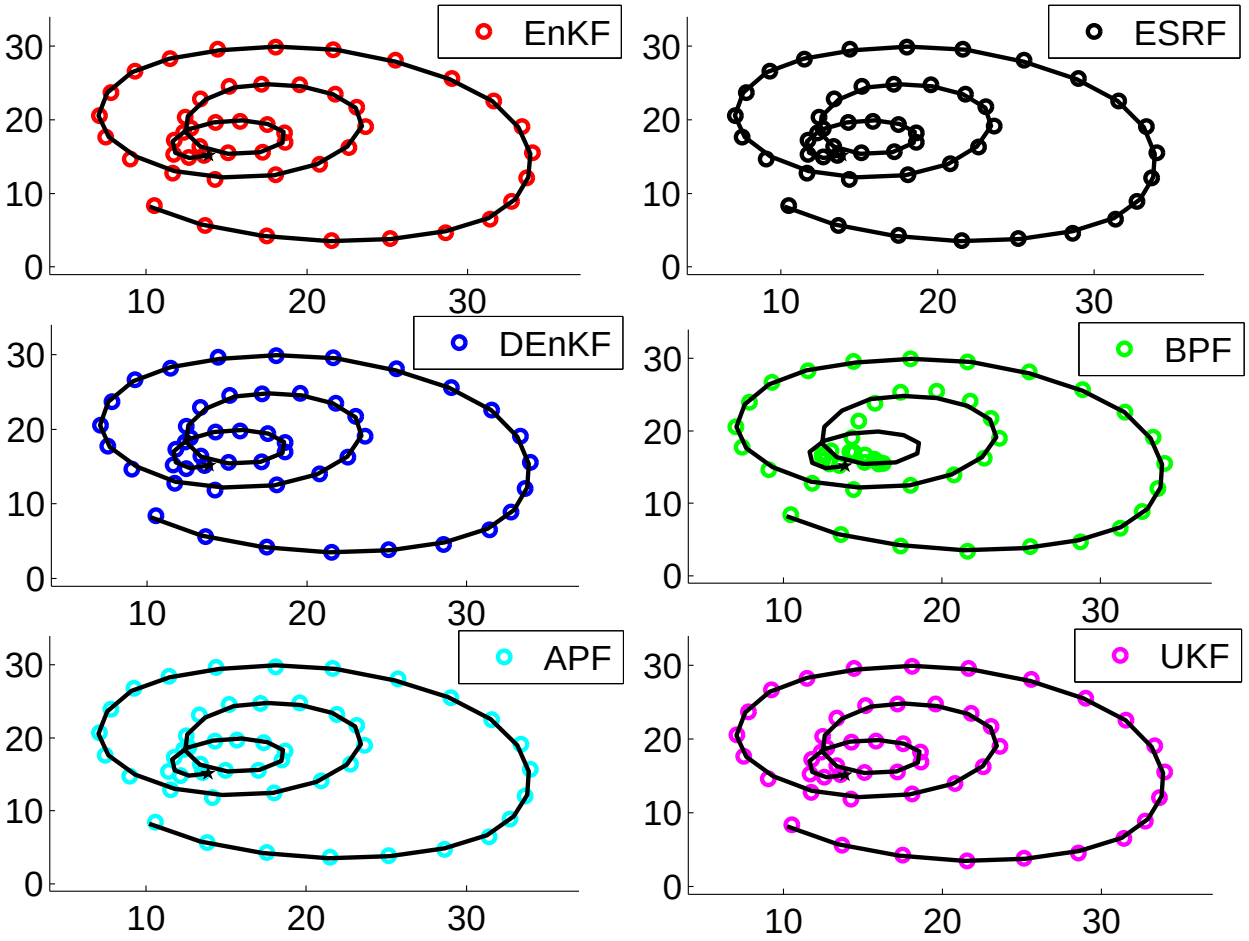

Fig. 4.3 Estimated target trajectories for the linear measurement model with $\sigma_\epsilon = 0.25$ and sample size of 100. The solid line indicates the true trajectory and the star represents the starting point.

Linear Measurement Model

Figure 4.3 shows the estimated target trajectories for a single run of the filters for the linear measurement model with $\sigma_\epsilon = 0.25$. Here, all the filters expect for the UKF use 100 samples. The unscented Kalman filtering approach does not use random samples, but uses a fixed number of deterministically chosen sigma points. The number of sigma points used by the UKF is defined by the dimensionality of the state vector. In the target tracking problem there are four state variables, hence the UKF uses nine sigma points throughout the simulations. It can be seen in the figure that the estimated tracks for the EnKFs and the UKF are closer to the actual target tracks as compared to the BPF and the APF.

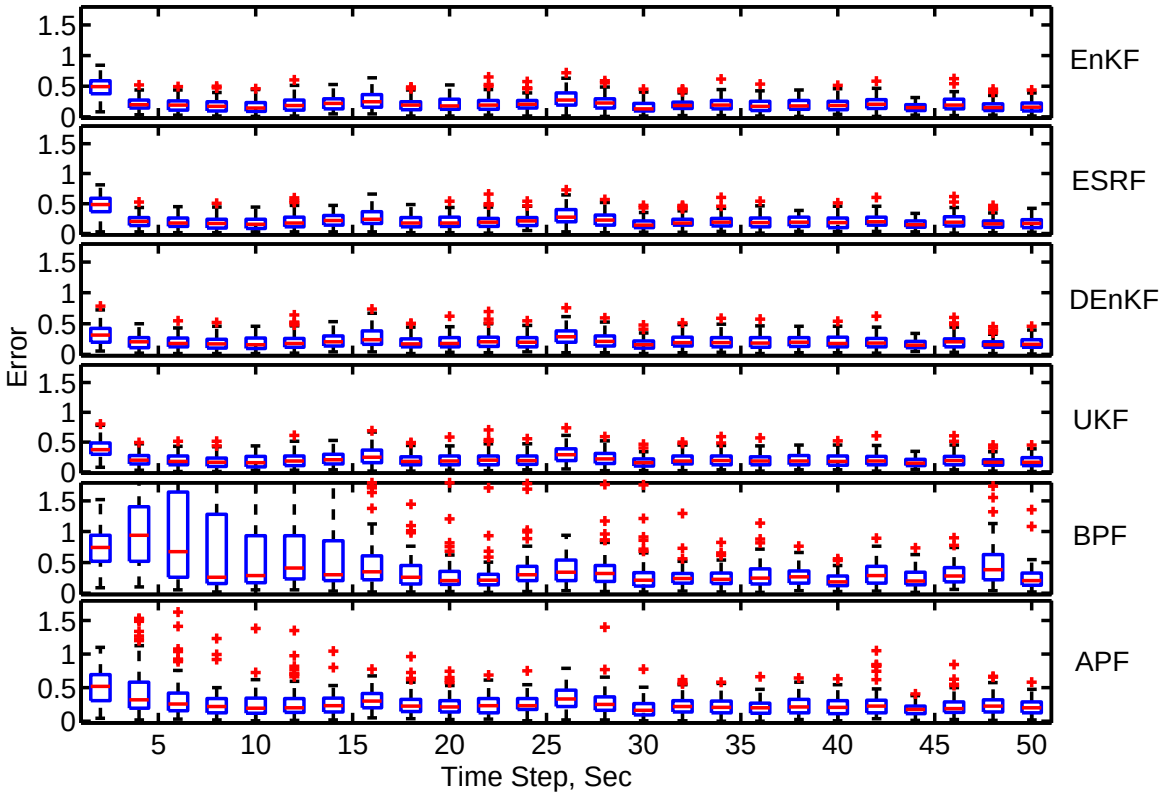

Fig. 4.4 Box-and-whisker plot of the RMS error over time for the linear model with $\sigma_\epsilon = 0.25$ and sample size of 100. Boxes cover 25-75 interquartile range, median is marked with a line, whiskers extend 1.5 times the interquartile range and + represents the outliers.

Figure 4.4 shows the box-and-whisker plot of the RMS error over time for 100 trials with $\sigma_\epsilon = 0.25$ and sample size of 100. The RMS error is plotted at every alternative time step. It can be seen in the figure that for all the filters the RMS error is slightly higher at the first few time steps and settles to a constant level. It can also be observed that the EnKFs and the UKF have a lower RMS error compared to BPF and the APF. Among the particle filtering methods the APF has a better performance. The EnKFs perform better compared to the particle filters because they have a lower requirement of the sample size.

In order to further investigate the effect of the sample size on the performance of these filters we compare their RMS error by varying the number of samples. Table 4.1 and Table 4.2 provide the performance comparison for the linear model with $\sigma_\epsilon = 0.25$ and $\sigma_\epsilon = 0.50$, respectively. The table contains the average RMS error $\pm$ standard deviation and the per-

Table 4.1 Performance comparison of the centralized filters for the linear measurement model with $\sigma_\epsilon = 0.25$.

	Ave. RMSE ± Std. Dev. (% track loss)				
Samples	1000	500	100	50	10
BPF	0.27 ± 0.09	0.30 ± 0.12	0.54 ± 0.29(14)	0.80 ± 0.37(25)	1.14 ± 0.45(94)
APF	0.25 ± 0.02	0.26 ± 0.02	0.31 ± 0.10	0.46 ± 0.28(1)	1.10 ± 0.39(42)
EnKF	0.23 ± 0.02	0.23 ± 0.02	0.24 ± 0.02	0.24 ± 0.02	0.28 ± 0.03
ESRF	0.23 ± 0.02	0.23 ± 0.02	0.24 ± 0.02	0.24 ± 0.02	0.26 ± 0.03
DEnKF	0.23 ± 0.02	0.23 ± 0.02	0.23 ± 0.02	0.23 ± 0.02	0.24 ± 0.02
UKF	0.23 ± 0.02	0.23 ± 0.02	0.23 ± 0.02	0.23 ± 0.02	0.23 ± 0.02

Table 4.2 Performance comparison of the centralized filters for the linear measurement model with $\sigma_\epsilon = 0.50$.

	Ave. RMSE ± Std. Dev. (% track loss)				
Samples	1000	500	100	50	10
BPF	0.44 ± 0.05	0.46 ± 0.07	0.71 ± 0.28(3)	0.91 ± 0.33(22)	1.31 ± 0.36(96)
APF	0.44 ± 0.03	0.44 ± 0.03	0.47 ± 0.06	0.55 ± 0.13	1.28 ± 0.32(47)
EnKF	0.43 ± 0.04	0.43 ± 0.04	0.43 ± 0.04	0.44 ± 0.04	0.42 ± 0.06
ESRF	0.43 ± 0.04	0.43 ± 0.04	0.43 ± 0.04	0.43 ± 0.04	0.47 ± 0.06
DEnKF	0.43 ± 0.04	0.43 ± 0.04	0.43 ± 0.04	0.43 ± 0.04	0.45 ± 0.05
UKF	0.43 ± 0.04	0.43 ± 0.04	0.43 ± 0.04	0.43 ± 0.04	0.43 ± 0.04

centage of lost tracks for all the filters at different sample sizes. It can be seen in both the tables that at the sample size of 1000 all the filters have comparable performance. As the sample size is decreased the RMS error for the BPF and the APF increases rapidly. On the contrary, the EnKFs have almost the same RMS error up to the sample size of 50 and only experience a slight increase in error at the sample size of 10. The EnKFs perform better for small sample sizes because they only approximate the first two moments of the posterior distribution. However, the particle filters approximate the complete posterior distributed and hence require a larger number of samples for good approximation. When the sample size is small the APF performs better than the BPF because it uses a superior sampling strategy and performs resampling twice in each iteration of the filter.

Table 4.3 shows the average running time of all the filters for the linear measurement

Table 4.3 Running time comparison of the centralized filters for the linear measurement model.

	Average running time (Sec)				
Samples	1000	500	100	50	10
BPF	0.49	0.22	0.05	0.03	0.01
APF	0.67	0.31	0.07	0.05	0.02
EnKF	0.19	0.10	0.03	0.02	0.02
ESRF	0.20	0.11	0.03	0.03	0.02
DEnKF	0.20	0.10	0.03	0.02	0.01
UKF	0.02	0.02	0.02	0.02	0.02

model. The running time is defined as the simulation time of the filter for 50 time steps. The running time is averaged over 100 trials and reported at different sample sizes. These simulations were performed in the Matlab software running on a system with the Intel Core-i7 processor (3.40 GHz) and 16 GB of RAM. It can be seen in the table that the running time of the EnKFs at a sample size of 50 is comparable to the UKF. Since, the EnKFs reach very close to their best performance at a sample size of 50, hence they have a similar computational cost compared to the UKF. However, the particle filters require at least a sample size of 1000 to give RMS error comparable to the EnKFs and UKF, hence they have a much higher computational cost.

Range Measurement Model

Figure 4.5 shows the estimated target trajectories for a single run of the filters for the range measurement model with $\sigma_\epsilon = 0.25$ and sample size of 100. The estimated tracks for the EnKF, ESRF and DEnKF are generated by using linearized approximation at each sample point.

In Figure 4.6, the box-and-whisker plot of the RMS error is presented for the range model with $\sigma_\epsilon = 0.25$ and sample size of 100. The results for the EnKF, ESRF and DEnKF are generated by using linearization at each sample point. Again, a similar trend to the linear model is observed in the box-and-whisker plot. All the filters have a slightly higher RMS error at the first few time steps and then settles to a constant level. The BPF and the APF have a higher RMS error because they have a higher requirement of the sample size and the APF performs relatively better than the BPF. The EnKFs have comparable performance to

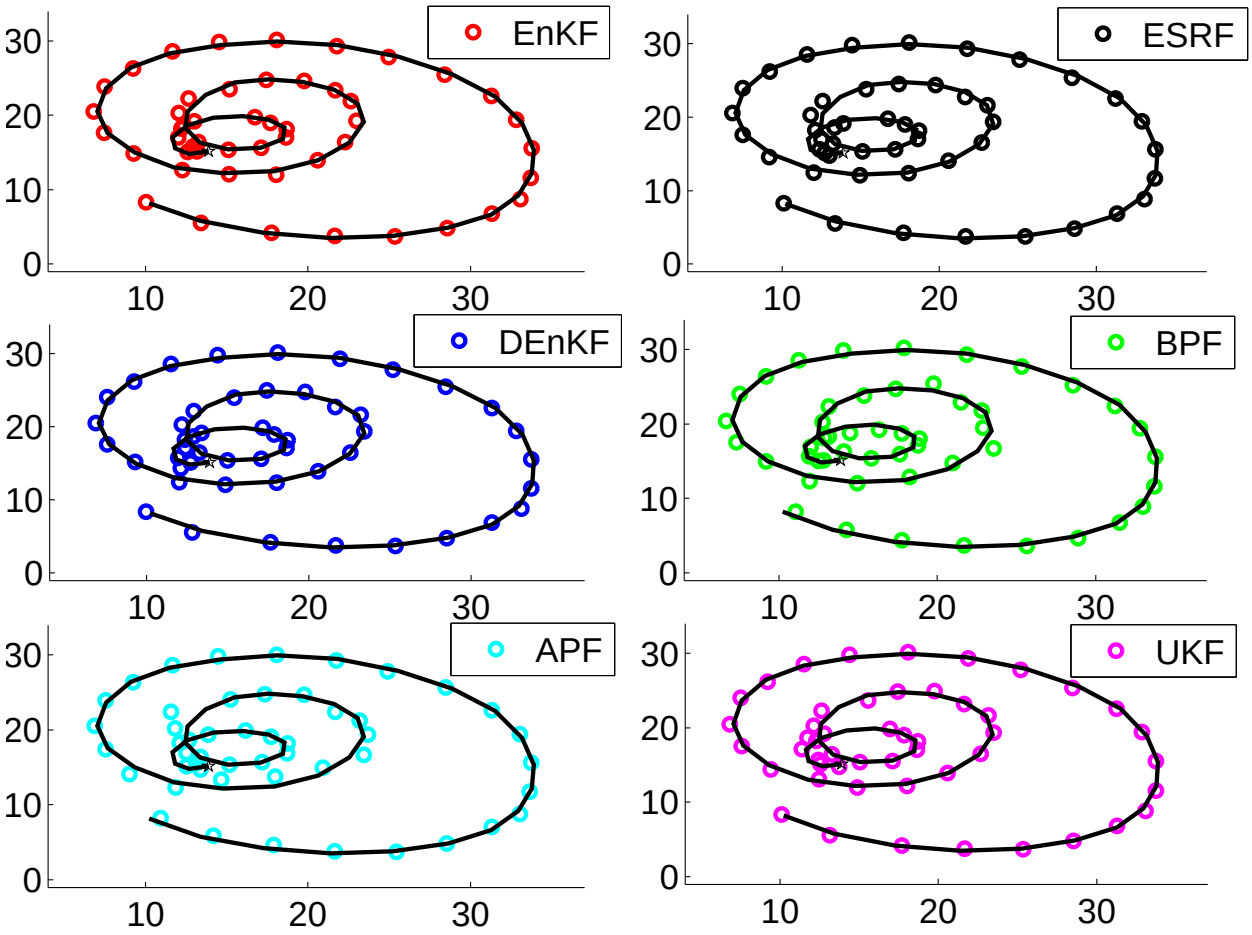

Fig. 4.5 Estimated target trajectories for the range measurement model with $\sigma_\epsilon = 0.25$ and sample size of 100. The EnKF, ESRF and DEnKF use linearization at each sample point.

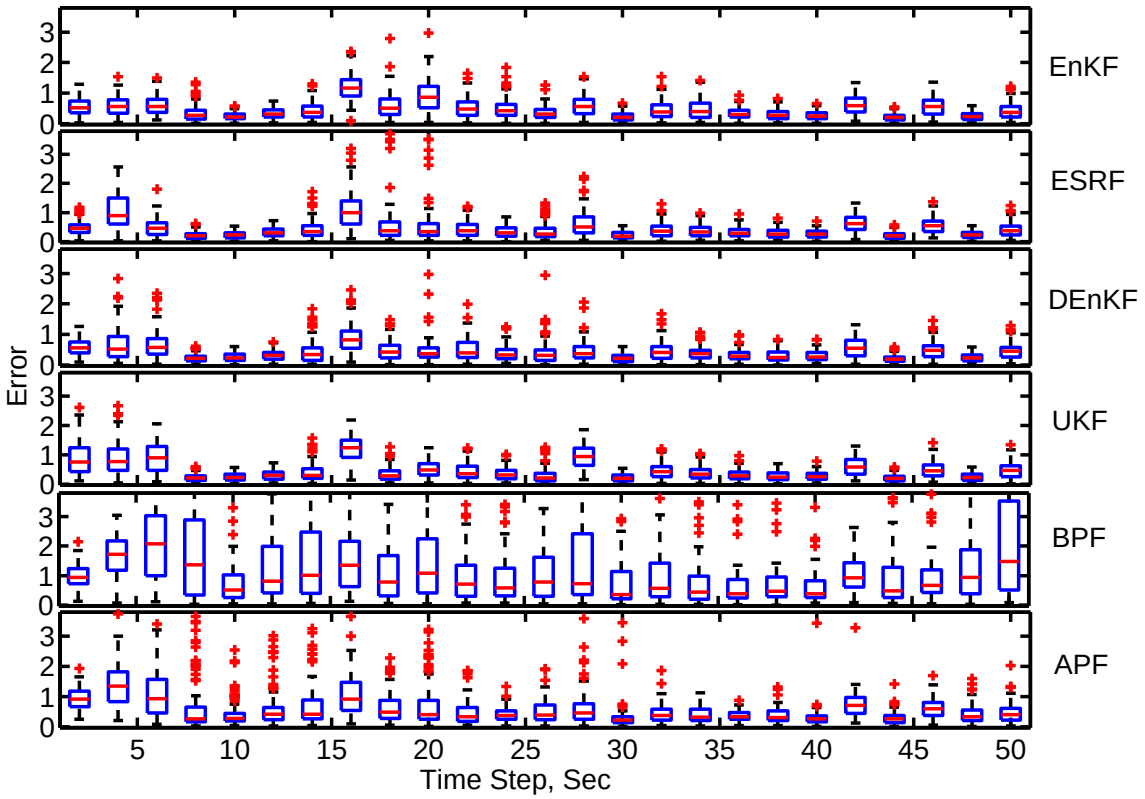

Fig. 4.6 Box-and-whisker plot of the RMS error over time for the linear model with $\sigma_\epsilon = 0.25$ and sample size of 100. The EnKF, ESRF and DEnKF use linearization at each sample point.

Table 4.4 Performance comparison of the centralized filters for the range measurement model with $\sigma_\epsilon = 0.25$.

Ave. RMSE ± Std. Dev. (% track loss)					
Samples	1000	500	100	50	10
BPF	0.58 ± 0.09	0.63 ± 0.12(1)	1.04 ± 0.39(30)	1.21 ± 0.35(55)	(100)
APF	0.56 ± 0.07	0.57 ± 0.08	0.73 ± 0.22(1)	0.95 ± 0.30(19)	1.75 ± 0.08(97)
UKF	0.60 ± 0.08	0.60 ± 0.08	0.60 ± 0.08	0.60 ± 0.08	0.60 ± 0.08
Linearization at each sample					
EnKF	0.58 ± 0.10	0.58 ± 0.10	0.59 ± 0.10	0.60 ± 0.12	0.71 ± 0.15
ESRF	0.58 ± 0.16	0.58 ± 0.17	0.59 ± 0.17	0.60 ± 0.16	0.65 ± 0.16
DEnKF	0.55 ± 0.14	0.55 ± 0.14	0.55 ± 0.14	0.56 ± 0.13	0.61 ± 0.13
Linearization at sample mean					
EnKF	(100)	(100)	1.37 ± 0.46(91)	1.22 ± 0.46(86)	0.98 ± 0.34(6)
ESRF	0.78 ± 0.25	0.78 ± 0.25	0.78 ± 0.26	0.77 ± 0.26	0.80 ± 0.27
DEnKF	0.71 ± 0.27	0.70 ± 0.27	0.70 ± 0.25	0.69 ± 0.22	0.73 ± 0.27
Pseudo measurement matrix representation					
EnKF	0.85 ± 0.16	0.84 ± 0.17	0.80 ± 0.19	0.78 ± 0.17	0.85 ± 0.21
ESRF	0.83 ± 0.24	0.83 ± 0.25	0.80 ± 0.22	0.78 ± 0.23	0.78 ± 0.25
DEnKF	0.80 ± 0.25	0.79 ± 0.24	0.78 ± 0.25	0.78 ± 0.24	0.76 ± 0.24
UKF [91]	0.64 ± 0.13	0.64 ± 0.13	0.64 ± 0.13	0.64 ± 0.13	0.64 ± 0.13

the UKF in terms of the median RMS error, but have a slightly larger number of outliers.

We conduct simulations to study the effect of the sample size and linearization strategy on the performance of the filters for the range model. Table 4.4 and Table 4.5 provide the performance comparison for the range model with $\sigma_\epsilon = 0.25$ and $\sigma_\epsilon = 0.50$, respectively. For the EnKFs, the performance results for each of the three linearization strategies are presented. We also include the results for the UKF that uses the pseudo measurement matrix based approximation [91]. The variation of the RMS error with the sample size shows a similar trend for all the filters as seen in the linear case. As sample size is decreased the BPF and the APF suffer from severe performance degradation while the EnKFs do not experience much change in their performance up to a sample size of 50. All three forms of the EnKF have a much lower RMS error when linearization is performed at each sample point compared to other linearization strategies. For this linearization strategy, the performance

Table 4.5 Performance comparison of the centralized filters for the range measurement model with $\sigma_\epsilon = 0.50$.

Ave. RMSE ± Std. Dev. (% track loss)					
Samples	1000	500	100	50	10
BPF	0.78 ± 0.09	0.82 ± 0.14	1.13 ± 0.27(25)	1.38 ± 0.34(59)	1.91 ± 0.00(99)
APF	0.77 ± 0.10	0.78 ± 0.11	0.92 ± 0.19(1)	1.14 ± 0.30(10)	1.90 ± 0.00(99)
UKF	0.85 ± 0.10	0.85 ± 0.10	0.85 ± 0.10	0.85 ± 0.10	0.85 ± 0.10
Linearization at each sample					
EnKF	0.78 ± 0.10	0.78 ± 0.10	0.80 ± 0.10	0.82 ± 0.12	1.02 ± 0.15
ESRF	0.77 ± 0.16	0.76 ± 0.17	0.77 ± 0.17	0.78 ± 0.16	0.90 ± 0.16
DEnKF	0.82 ± 0.81	0.81 ± 0.82	0.82 ± 0.14	0.82 ± 0.13	0.90 ± 0.13
Linearization at mean					
EnKF	1.67 ± 0.00(99)	1.50 ± 0.23(95)	1.57 ± 0.28(88)	1.47 ± 0.35(63)	1.15 ± 0.27(5)
ESRF	0.94 ± 0.18(1)	0.94 ± 0.20	0.97 ± 0.24(1)	0.94 ± 0.19	1.00 ± 0.26(1)
DEnKF	0.97 ± 0.24	0.98 ± 0.23	0.96 ± 0.22	1.00 ± 0.25	1.03 ± 0.27
Pseudo measurement matrix representation					
EnKF	1.02 ± 0.18	1.00 ± 0.16	0.98 ± 0.16	0.96 ± 0.15	1.12 ± 0.26(4)
ESRF	0.98 ± 0.19	0.97 ± 0.19	0.97 ± 0.19	0.95 ± 0.20	1.01 ± 0.19
DEnKF	1.07 ± 0.23	1.06 ± 0.22	1.04 ± 0.23	1.03 ± 0.23	1.08 ± 0.22
UKF [91]	0.87 ± 0.12	0.87 ± 0.12	0.87 ± 0.12	0.87 ± 0.12	0.87 ± 0.12

of the EnKFs is comparable to the best performance of the BPF, APF and the UKF. For the ESRF and the DEnKF, the sample mean is updated separately and perturbations are added to generate the updated samples. The analysis scheme in these filters is designed such that the perturbations are zero mean. Both the linearization at the sample mean and the pseudo measurement matrix representation strategies generate an approximation of the measurement model corresponding to the mean of the samples, thus the error in the update of the mean is mostly due to the bad estimates of the error covariances. Hence, the ESRF and the DEnKF perform relatively better for these linearization strategies as compared to the EnKF. In the EnKF with linearization at the sample mean, each sample is updated individually using the linearized approximation corresponding to the mean, thus the update causes all the samples to move in the direction defined by the sample mean. This causes the samples on the opposite side to the mean to move further away from the true target location which severely degrades the performance of the filter. We demonstrate this phenomenon by plotting the samples and their motion during the update step in the EnKF for a sample size of 200. Figure 4.7, Figure 4.8 and Figure 4.9 show the samples and their movement at time step 2, 4 and 6, respectively. It is evident in all the figures that all the samples move in the direction defined by the mean and the samples that are opposite to the mean move further away from the true target location. This introduces a large error in the calculation of the sample mean. The results in Table 4.4 and Table 4.5 show that for this linearization strategy the performance of the EnKF degrades with increase in the sample size and filter divergence is observed at higher sample sizes. This is due to the fact that all the samples move in the direction defined by the sample mean, hence increasing the sample size results in more samples going in the wrong direction and consequently, a higher error. The EnKF does not experience such a severe performance degradation when the pseudo measurement matrix representation is used, however the error in this case is much higher in comparison to the linearization at each sample point. The pseudo measurement matrix representation works better with the UKF and its RMS error is relatively close to the RMS error achieved by the BPF and APF at a sample size of 1000.

Since all three forms of the EnKF have a much better performance when linearization is performed at each sample point compared to the other linearization strategies, thus we only consider the EnKFs with linearization at each sample for the running time analysis. Table 4.6 shows the average running time of all the filters for the range measurement model. The running time corresponds to 50 estimation steps. These simulations were performed in

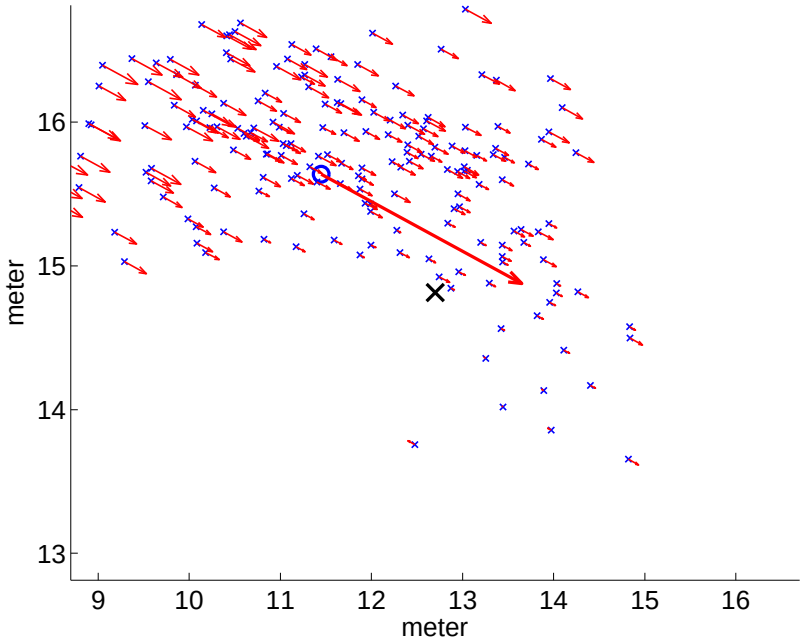

Fig. 4.7 Samples and their movement during update step in the EnKF (linearization at sample mean) for a sample size of 200 at time step 2. The small crosses represent the samples, circle represents the sample mean and the large cross indicates the true location of the target. The small arrows indicate the direction in which the samples move and the large arrow represents the direction of movement of the sample mean.

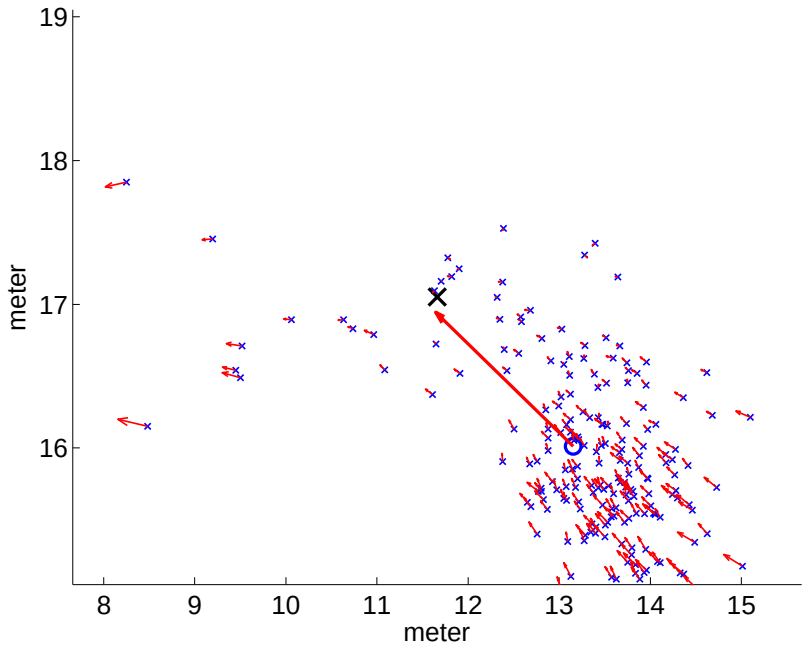

Fig. 4.8 Samples and their movement during update step in the EnKF (linearization at sample mean) for a sample size of 200 at time step 4.

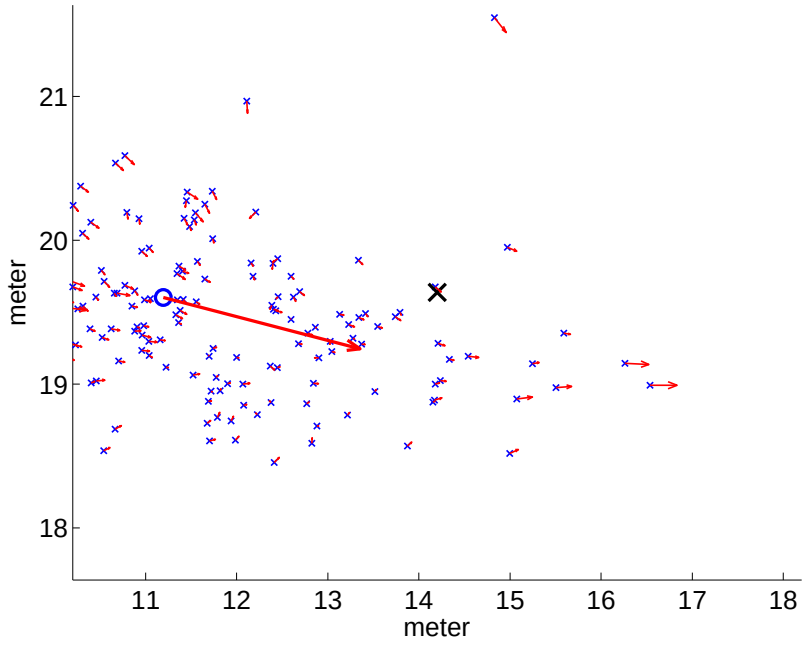

Fig. 4.9 Samples and their movement during update step in the EnKF (linearization at sample mean) for a sample size of 200 at time step 6.

Table 4.6 Running time comparison of the centralized filters for the range measurement model. The EnKF, ESRF and DEnKF use linearization at each sample.

Average running time (Sec)					
Samples	1000	500	100	50	10
BPF	0.50	0.23	0.06	0.04	0.03
APF	0.71	0.34	0.10	0.07	0.04
EnKF	0.46	0.25	0.09	0.07	0.05
ESRF	4.01	2.04	0.46	0.26	0.10
DEnKF	0.47	0.26	0.10	0.08	0.06
UKF	0.02	0.02	0.02	0.02	0.02

Matlab and all the filter implementations except for the ESRF were vectorized. The ESRF requires the evaluation of the matrix square root corresponding to each sample and we were unable to vectorize this matrix operation. Hence, the reported running time for the ESRF is substantially higher as compared to all the other filters. In case of vectorized matrix square root computation the running time of the ESRF is expected to be slightly higher than the DEnKF. It can be seen in the table that the running time of the EnKF and the DEnKF at a sample size of 50 is comparable to the UKF. However, the BPF and the APF have a higher computational overhead because of their requirement of a larger sample size.

Discussion

In a centralized tracking problem with non-linear dynamics and a linear measurement model the EnKF, ESRF and the DEnKF give comparable performance to the particle filters, but have a much lower computational cost. They closely match the UKF in both the estimation accuracy and the computational cost. Their computationally complexity is low due to the fact that they require a much lower sample size as compared to the BPF and the APF. The results show that a sample size of 50 is sufficient for the EnKFs to attain their best performance and a very small performance gain is achieved by further increasing the sample size. In case of the non-linear measurement model, linearization at each sample point appears to be the best linearization strategy for the EnKFs. In the cases of the linearization at the sample mean and the pseudo measurement matrix representation, the RMS error of the EnKFs is significantly higher. For linearization at each sample point, the EnKFs are able

to achieve performance comparable to performance of the BPF and the APF. Similar to the linear case, the EnKFs reach very close to their best performance at a sample size of 50. Hence, they have a much lower computational cost and similar performance compared to the BPF and the APF for non-linear measurement models.

4.2.4 Performance Comparison of Distributed Filters

This section provides the performance comparison of the distributed filters in terms of the communication efficiency. We compare the proposed distributed filtering scheme, namely the EnKF, ESRF and DEnKF with the existing state-of-the-art distributed schemes. The proposed distributed schemes use the randomized gossip algorithm [11] to perform gossip because most of the existing distributed filtering schemes have reported results based on this algorithm. In future work, one could investigate how different accelerated gossip procedures affect the proposed and comparative schemes. We consider the following methods for comparison: the Gaussian approximation (GA) distributed particle filter [6], the sequential importance resampling (SIR) version of the likelihood consensus (LC) distributed particle filter [7], the set membership (SM) distributed particle filter [8], the top-m selective gossip (Top-m) distributed particle filter [9], and the distributed unscented Kalman filter (UKF) [78]. The distributed filtering schemes [81, 82] could not be included in this comparative study because these methods are developed specifically for the bearings-only measurement model and do not have the ability to incorporate other measurement models. These comparative methods are discussed in Section 2.4. For the non-linear measurement models, the proposed distributed filtering schemes use linearization at the each sample point because it gives better performance compared to the other linearization strategies. For the likelihood consensus approach, we use a Taylor series approximation of degree one. In the top-m selective gossip approach, we choose $m = N/4$, where N represents the sample size. In the set membership scheme, we set the oversampling parameter to 10 which is the value used in the original paper. The distributed unscented Kalman filtering scheme models the posterior distribution as a Gaussian mixture. The communication cost of this scheme increases linearly with the number of mixture components, hence we use only one mixture component to minimize its communication cost. We perform a series of simulations to compare the performance of the distributed filters at different communication costs. In each simulation all the filters transmit approximately the same number of scalars in the network. In general,

the communication cost of the distributed filters depends on the sample size and the number of gossip iterations, hence multiple filter settings result in the same communication cost. For the comparative filters, we choose the sample size and the number of gossip iteration for a certain communication cost, such that their performance is optimized. However, we do not perform any optimization for the proposed filters and run them with default parameters. Specifically, we choose a moderate sample size of 50 or less because the EnKFs do not give any performance gain when the sample size is increased beyond 50. In the following sections, we present the performance results for the linear, range and RF tomography measurement models.

Linear Measurement Model

Table 4.7 provides the performance comparison of the distributed filters for the linear model with $\sigma_\epsilon = 0.25$. The table contains the average RMS error $\pm$ standard deviation and the percentage of lost tracks for the distributed filters at different communication costs. The communication cost is measured in terms of the scalars transmitted per sensor per time-step. Table B.1 gives information about the filter parameters that resulted in the presented communication costs. We have also included the performance results for the centralized bootstrap particle filter (BPF) with a sample size of 2000 to serve as an approximate lower bound on the performance. The distributed UKF scheme [78] employs pseudo measurement matrix approximation in its analysis scheme. We also include a variant of this scheme in which the pseudo measurement matrix is replaced by the actual linear measurement matrix. This new scheme is denoted by UKF* in Table 4.7. For some filters we were unable to find a working point corresponding to a particular communication cost, hence we have marked these cases with a dash. The results show that at a high communication cost all the distributed filters have performance close to the centralized BPF. As the communication cost is decreased all the filters except for the ESRF and DEnKF experience considerable performance degradation. The LC and the UKF schemes gossip on low dimensional data, but have a much higher demand of gossip iterations. These schemes are more sensitive to the gossip errors and require the local estimates at the sensor nodes to be closer to the true average. Among these schemes, the LC approach has a higher sensitivity. The SM and the Top-m schemes gossip on the high dimensional particle weights which increases their communication overhead. Since the SM scheme performs oversampling of the particles it can

Table 4.7 Performance comparison of the distributed filters for the linear measurement model with $\sigma_\epsilon = 0.25$.

Ave. RMSE ± Std. Dev. (% track loss)			
Scalars	8000	480	200
GA [6]	0.26 ± 0.02	0.26 ± 0.02	0.28 ± 0.02(1)
LC [7]	0.24 ± 0.02	0.24 ± 0.02	0.84 ± 0.14
SM [8]	0.29 ± 0.10	0.45 ± 0.02	0.49 ± 0.17(1)
Top-m [9]	0.24 ± 0.02	-	-
UKF [78]	0.24 ± 0.02	0.37 ± 0.20	-
UKF* [78]	0.24 ± 0.02	0.26 ± 0.04	0.82 ± 0.43(12)
EnKF	0.24 ± 0.02	0.34 ± 0.04	-
ESRF	0.24 ± 0.02	0.24 ± 0.02	0.27 ± 0.02
DEnKF	0.24 ± 0.02	0.24 ± 0.02	0.28 ± 0.02
Cent. BPF	0.24 ± 0.02	0.24 ± 0.02	0.24 ± 0.02

still work at lower communication costs by using a small sample size. The GA scheme is more balanced in terms of the data dimensionality and the demand of gossip. However, it is more sensitive to the filter settings, e.g., level of regularization noise. In the low communication regime, the GA scheme performs well most of the time, but generates occasional lost tracks. The ESRF and DEnKF gossip on low dimensional variables and also have a lower demand of gossip iterations, hence they perform better in a low communication regime. The EnKF scheme gossips on higher dimensional variables as compared to the ESRF and DEnKF, therefore it has a higher communication overhead. The UKF scheme performs comparatively better in the low communication regime when the pseudo measurement matrix is replaced by the actual measurement matrix, but still has a substantially higher error compared to the ESRF and DEnKF.

Range Measurement Model

Table 4.8 presents the performance comparison for the range model with $\sigma_\epsilon = 0.25$. Table B.2 gives information about the filter parameters that resulted in the presented communication costs. Here, the EnKF, ESRF and DEnKF use linearization at each sample point. Hence, these schemes are required to gossip on all the instances of the local variables corresponding to each sample. This substantially increases the data dimensionality and consequently, the

Table 4.8 Performance comparison of the distributed filters for the range measurement model with $\sigma_\epsilon = 0.25$.

	Ave. RMSE ± Std. Dev. (% track loss)		
Scalars	16000	8000	800
GA [6]	0.55 ± 0.11(1)	0.57 ± 0.08(5)	0.59 ± 0.12(3)
LC [7]	0.87 ± 0.20(1)	0.86 ± 0.18(2)	0.86 ± 0.20(1)
SM [8]	0.61 ± 0.10	0.71 ± 0.18(3)	0.83 ± 0.21(5)
Top-m [9]	0.58 ± 0.08	0.61 ± 0.11	-
UKF [78]	0.64 ± 0.14	0.64 ± 0.14	0.73 ± 0.14
EnKF	0.78 ± 0.16	0.80 ± 0.18	-
ESRF	0.67 ± 0.22	0.72 ± 0.26	-
DEnKF	0.69 ± 0.25(2)	0.72 ± 0.29(1)	-
Cent. BPF	0.57 ± 0.08	0.57 ± 0.08	0.57 ± 0.08

communication overhead of these filters compared to the linear case. All the comparative filters have the same communication overhead at a particular parameter setting for both the linear and range measurement models. However, all the filters require more samples and/or more gossip iterations to give good performance as compared to the linear model. This is because the range tracking problem is a more difficult problem. The results show that the UKF performs better in a low communication regime. This is because the UKF scheme gossips on a low dimensional data and a communication cost of $1.0e+06$ allows it to perform sufficient gossip to achieve good performance. Again, the GA scheme has good performance most of the times, but suffers from occasional lost tracks.

RF Tomography Measurement Model

In the RF tomography tracking problem, we assume that every sensor node is aware of the measurement model parameters of the other nodes. However, the sensor nodes cannot transmit raw measurements because they have high dimensionality. In this case, the EnKF, ESRF, DEnKF and UKF only need to gossip on the local variables that involve the measurements. The EnKFs can perform linearized approximations and compute the corresponding statistics locally at every node. On the other hand, there is no change in the dimensionality to the data on which the particle filtering methods gossip. Hence, EnKF, ESRF, DEnKF and UKF gossip on a much lower dimensional data as compared to the distributed particle filter-

ing methods. Table 4.9 provides the performance comparison for the RF tomography model with $\sigma_\epsilon = 0.50$, $\sigma_\lambda = 0.05$ and $\phi = 5$. Table B.3 gives information about the filter parameters that resulted in the presented communication costs. The results show that the EnKF does not work under any parameter setting. This is because the EnKF involves the perturbation of measurements that introduce additional sampling errors. As the measurement dimensionality increases the sampling errors also increase, and consequently result in filter divergence for high dimensional measurements. At a high communication cost $(2.0e + 07)$, the performance of the ESRF, DEnKF and UKF is essentially the same as their centralized versions. In the case of the range model, it is observed that the centralized ESRF, DEnKF and UKF have comparable performance to the BPF. However, here we observe a visible performance gap with the centralized BPF doing much better. This is because the range model is only mildly non-linear, but the RF tomography model is highly non-linear. As the non-linearity in the measurement model increases the posterior distribution develops more non-Gaussian components and the parametrization of the posterior distribution by mean and covariance results in a less accurate estimate. The results show that the UKF suffers more performance degradation due to the increased non-linearity as compared to the ESRF and the DEnKF. In the low communication regime, the ESRF and DEnKF perform better than all the other filters, however the DEnKF does suffer from occasional lost tracks. The ESRF and DEnKF perform better because they need to gossip on a very low dimensional data compared to the distributed particle filtering methods. The UKF also gossips on a low dimensional data, but has a much higher demand of gossip. Hence, its performance degrades significantly at lower communication costs.

Discussion

In the distributed tracking problem with non-linear dynamics and a linear measurement model, two of the proposed distributed schemes, namely, the ESRF and the DEnKF give comparable performance to the state-of-the-art distributed filters, but have a much higher communication efficiency. This is because these schemes gossip on low dimensional data and have a lower requirement of gossip compared to the existing methods. However, in the case of the range model with linearization at each sample point the data dimensionality increase which causes a significant performance reduction at low communication costs. In the case of the RF tomography model with all the model parameters known at every sensor node

Table 4.9 Performance comparison of the distributed filters for the RF tomography measurement model with $\sigma_\epsilon = 0.50$, $\sigma_\lambda = 0.05$ and $\phi = 5$.

	Ave. RMSE ± Std. Dev. (% track loss)		
Scalars	16667	500	350
GA [6]	0.20 ± 0.01(6)	0.18 ± 0.01(6)	0.18 ± 0.01(8)
LC [7]	0.23 ± 0.15(4)	-	-
SM [8]	0.14 ± 0.08(4)	-	-
Top-m [9]	0.12 ± 0.04	-	-
UKF [78]	0.18 ± 0.14(2)	0.59 ± 0.49(62)	0.60 ± 0.50(81)
EnKF	-	-	-
ESRF	0.20 ± 0.04	0.21 ± 0.06	0.20 ± 0.07
DEnKF	0.23 ± 0.09	0.23 ± 0.08(1)	0.20 ± 0.08(2)
Cent. BPF	0.10 ± 0.01	0.10 ± 0.01	0.10 ± 0.01

the ESRF and DEnKF outperform the existing state-of-the-art schemes. In this case, the ESRF and DEnKF schemes allow the sensor nodes to compute linearized approximations and evaluate the corresponding statistics locally. This results in a significant reduction in the dimensionality of the data on which the sensor nodes gossip and hence, a low communication overhead.

4.3 Summary

In this chapter, we provided a performance comparison of the proposed distributed schemes and the existing state-of-the-art distributed filters for a target tracking problem. In the centralized setting, we compared the performance of the EnKFs to the UKF and the particle filtering methods. The results suggested that the EnKFs provide comparable performance to the UKF and the particle filters, but have a significantly lower computational cost compared to the particle filters. The EnKFs and the UKF have comparable computational overhead. This is due to the fact that the EnKFs require a relatively small sample size to achieve their best performance. The results for distributed tracking showed that the proposed schemes outperform the existing state-of-the-art distributed filtering schemes in terms of communication efficiency for linear measurement models and/or high dimensional measurements. In both the cases the state dynamics are non-linear. The proposed methods have a better communication efficiency because they gossip on lower dimensional data and require fewer

gossip iterations to achieve performance comparable to the existing distributed schemes.

Chapter 5

Conclusions

In this thesis, we addressed the problem of distributed estimation in a wireless sensor networks (WSNs). The existing distributed filtering schemes either achieve good estimation accuracy at a high communication cost, or have low communication overhead, but do not work well for non-linear and non-Gaussian problems. In a resource constrained and remotely deployed wireless sensor network, the communication overhead of the distributed filtering scheme becomes critical and dictates the network lifetime. Motivated by this fact, we proposed three distributed filtering schemes with the goal of reducing the communication overhead.

The proposed distributed schemes are based on three forms of the ensemble Kalman filter (EnKF). The EnKF uses a sample based approach to approximate the mean and covariance of the posterior distribution. It uses the state dynamics to propagate the samples which allows it to capture the non-linearities in the state dynamics. However, the EnKF analysis equations still require the measurement model to be linear. We also presented three linearization strategies to incorporate non-linear measurement models into the EnKF analysis.

The main challenge in developing a distributed approximation of the EnKF lies in the formulation of the distributed analysis mechanism. The analysis step requires complete knowledge of the measurements and measurement model parameters, while the sensor nodes only have access to their local measurement and model parameters. We overcome this difficulty by expressing the analysis equation in an alternative information form. This allows the sensor nodes to gossip and reach consensus on a set of statistics that are sufficient to

implement the analysis equations locally.

In a centralized setting, we compared the performance of the three ensemble Kalman filtering schemes, namely the ensemble Kalman filter (EnKF), ensemble square root filter (ESRF) and the deterministic ensemble Kalman filter (DEnKF) with the unscented Kalman filter (UKF) and the particle filtering schemes. The results showed that the EnKFs have comparable performance to the particle filters, but have a much lower computational cost. Also, the EnKFs and the UKF have comparable performance and computational cost. The EnKFs have a much lower computational cost compared to the particle filters because they require a relatively small sample size to provide a good approximation of the mean and covariance to the posterior distribution. In the case of non-linear measurements, linearization at the each sample gives a better performance compared to the other linearization strategies.

We also presented a performance comparison of the proposed schemes and the existing state-of-the-art distributed filtering schemes. The results suggested that the proposed schemes give comparable performance to the existing schemes while significantly reducing the communication overhead in two scenarios, a) linear measurement model with non-linear state dynamics, and b) high dimensional measurements (model parameters known everywhere in the network) with non-linear measurement model and state dynamics. The proposed schemes have a better communication efficiency because they gossip on lower dimensional data and have a lower demand of gossip.

In this thesis, the performance of the filtering schemes is compared for a single target tracking problem. For future work, these schemes can be compared for a multi-target tracking problem which has a higher state dimensionality. The increased state dimensionality affects the Kalman filtering and particle filtering methods in different ways. For Kalman filtering methods, it increases the dimensionality of the local variables on which gossip is performed. Particle filtering methods require much larger sample sizes to achieve the same accuracy when tracking higher dimensional states. The proposed distributed ensemble Kalman filtering schemes gossip equally on all elements of the exchanged vectors and matrices. It may be the case that some elements of the vectors or matrices, e.g. the diagonal elements of the matrices, may have a higher impact on the estimation accuracy. Hence, the communication overhead of the proposed schemes may be further reduced by gossiping on selective elements.

Appendix A

Incorporating Non-linear Measurement Models in EnKFs

A.1 Linearization Strategies for EnKFs

The ensemble Kalman filters are designed for non-linear dynamical models, but require the measurement model to be linear. In this appendix, we present three methods for incorporating the non-linear measurement models into the EnKF analysis equations. We present these methods for all three forms of the EnKF.

A.1.1 Linearization at Mean

This linearization approach is very similar to the extended Kalman filter linearization approach. The measurement model is linearized at the mean of the forecast samples $\hat{\mathbf{x}}^f$ and the same linearized approximation is used in the analysis equation for all the samples. This approach might be useful when the forecast sample covariance is small and all the samples are close to the mean. The linearized approximation is defined as

$$\mathbf{H} = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}\Big|_{\hat{\mathbf{x}}^f}. \tag{A.1}$$

This method requires a common Kalman gain for all the samples which can be computed as follows

$$\mathbf{K} = \mathbf{P}^f\mathbf{H}^T(\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R})^{-1}. \tag{A.2}$$

The EnKF analysis equation for each sample can then be written as follows

$$\mathbf{x}^a_{(i)} = \mathbf{x}^f_{(i)} + \mathbf{K}(\mathbf{y}_{(i)} - h(\mathbf{x}^f_{(i)})). \tag{A.3}$$

Similarly, the analysis equations for the ESRF can be written as

$$\hat{\mathbf{x}}^a = \hat{\mathbf{x}}^f + \mathbf{K}(\mathbf{y} - h(\hat{\mathbf{x}}^f)), \tag{A.4}$$

$$\mathbf{T} = (\mathbf{I} - \mathbf{K}\mathbf{H})^{1/2}, \tag{A.5}$$

$$\mathbf{x}^a_{(i)} = \hat{\mathbf{x}}^a + \mathbf{T}(\mathbf{x}^f_{(i)} - \hat{\mathbf{x}}^f)\ . \tag{A.6}$$

The DEnKF uses the same analysis equations as the ESRF except for the equation to compute the transformation matrix which is given by

$$\mathbf{T} = \mathbf{I} - \frac{1}{2}\mathbf{K}\mathbf{H}. \tag{A.7}$$

A.1.2 Pseudo Measurement Matrix Representation

In this linearization approach, we adopt the use of the pseudo measurement matrix as a linearized representation of a non-linear model for the EnKFs. The pseudo measurement matrix has been defined and used for the unscented Kalman filter in [79, 91]. First, we define the predicted cross-covariance between the state variables and the measurements as follows

$$\hat{\mathbf{y}} = \frac{1}{N}\sum_{i=1}^{N} h(\mathbf{x}^f_{(i)}), \tag{A.8}$$

$$\widetilde{\mathbf{P}} = \frac{1}{N-1}\sum_{i=1}^{N} (\mathbf{x}^f_{(i)} - \hat{\mathbf{x}}^f)(h(\mathbf{x}^f_{(i)}) - \hat{\mathbf{y}})^T, \tag{A.9}$$

where N is the number of samples.
The pseudo measurement matrix can then be defined as

$$\mathbf{H} = (\widetilde{\mathbf{P}})^T(\mathbf{P}^f)^{-1}. \tag{A.10}$$

This pseudo measurement matrix serves as a common linearization and can be used to update all the samples in the same way as in the linearization at mean approach.

A.1.3 Linearization at Each Sample

A more accurate way of adopting the extened Kalman filter linearization mechanism in EnKFs is to perform linearization at each sample and use this sample specific linearized approximation in the analysis equations. This is a more accurate way to capturing the non-linearities in the model, but has a much higher computational cost. The linearized model for the sample $\mathbf{x}^f_{(i)}$ can be evaluated as

$$\mathbf{H}_{(i)} = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}\Big|_{\mathbf{x}^f_{(i)}}. \tag{A.11}$$

Hence, there is a requirement to compute an instance of the Kalman gain corresponding to each sample. The Kalman gain for i^{th} sample is given by

$$\mathbf{K}_{(i)} = \mathbf{P}^f\mathbf{H}^T_{(i)}(\mathbf{H}_{(i)}\mathbf{P}^f\mathbf{H}^T_{(i)} + \mathbf{R})^{-1}. \tag{A.12}$$

The analysis equation for the EnKF becomes

$$\mathbf{x}^a_{(i)} = \mathbf{x}^f_{(i)} + \mathbf{K}_{(i)}(\mathbf{y}_{(i)} - h(\mathbf{x}^f_{(i)})). \tag{A.13}$$

For this linearization methodology, the ESRF and the DEnKF use the same mean update equation (A.4) as in the previous linearization approach. However, an instance of the transformation matrix corresponding to each sample needs to be computed. In the ESRF scheme, the transformation matrix for i^{th} sample is given by

$$\mathbf{T}_{(i)} = (\mathbf{I} - \mathbf{K}_{(i)}\mathbf{H}_{(i)})^{1/2}. \tag{A.14}$$

Similarly, in the DEnKF scheme, the transformation matrix for the i^{th} sample is evaluated as

$$\mathbf{T}_{(i)} = \mathbf{I} - \frac{1}{2}\mathbf{K}_{(i)}\mathbf{H}_{(i)}. \tag{A.15}$$

A.2 Linearized Approximations

In this section, we present the linearized approximations of the range and RF tomography measurement model. The linearized approximation requires the evaluation of the partial derivatives of the model function with respect to each of the state variables. In the following sections, the expressions for the partial derivatives are presented for the range and RF tomography model.

A.2.1 Range Model

The range model function for the sensor node v is defined as follows

$$h^v(\mathbf{x}) = \sqrt{(x_1 - s_x^v)^2 + (x_2 - s_y^v)^2}, \tag{A.16}$$

where x_1 and x_2 represent the position of target along the x-axis and y-axis and s_x^v and s_y^v represent the position of sensor v along the x-axis and y-axis, respectively. The partial derivative of the model function with respect to x_1 (x-component of target location) is given by

$$\frac{\partial}{\partial x_1}\left(h^v(\mathbf{x})\right) = \frac{\partial}{\partial x_1}\left(\left[(x_1 - s_x^v)^2 + (x_2 - s_y^v)^2\right]^{1/2}\right), \tag{A.17}$$

$$= \left[(x_1 - s_x^v)^2 + (x_2 - s_y^v)^2\right]^{-1/2}(x_1 - s_x^v). \tag{A.18}$$

Similarly, the partial derivative of the model function with respect to x_2 (y-component of target location) is given by

$$\frac{\partial}{\partial x_2}\left(h^v(\mathbf{x})\right) = \left[(x_1 - s_x^v)^2 + (x_2 - s_y^v)^2\right]^{-1/2}(x_2 - s_y^v). \tag{A.19}$$

Since the model function does not involve the state variables x_3 and x_4, therefore the partial derivatives with respect to these variables are zero.

A.2.2 RF Tomography Model

The RF tomography measurements are defined in terms of the attenuation experienced between pairs of sensors. Hence, the linearization of the model requires the computation of the partial derivatives of the attenuation function with respect to each of the state variables. The attenuation function for the sensor nodes u and v is defined as

$$A^{(u,v)}(\mathbf{x}) = \phi \exp\left(-\frac{\lambda^{(u,v)}(\mathbf{x})^2}{2\sigma_\lambda^2}\right). \tag{A.20}$$

The partial derivative of the attenuation function $A^{(u,v)}(\mathbf{x})$ with respect to the i^{th} component of the state variable is given by

$$\frac{\partial}{\partial x_i}(A^{(u,v)}(\mathbf{x})) = \frac{\partial}{\partial x_i}\left(\phi \exp(-\frac{\lambda^{(u,v)}(\mathbf{x})^2}{2\sigma_\lambda^2})\right), \tag{A.21}$$

$$= \phi(-\frac{\lambda^{(u,v)}(\mathbf{x})}{\sigma_\lambda^2}) \exp(-\frac{\lambda^{(u,v)}(\mathbf{x})^2}{2\sigma_\lambda^2}) \times \frac{\partial}{\partial x_i}\left(\lambda^{(u,v)}(\mathbf{x})\right). \tag{A.22}$$

The function $\lambda^{(u,v)}(\mathbf{x})$ is defined as follows

$$\lambda^{(u,v)}(\mathbf{x}) = d^u(\mathbf{x}) + d^v(\mathbf{x}) - d^{(u,v)}, \tag{A.23}$$

$$= \sqrt{(x_1 - s_x^u)^2 + (x_2 - s_y^u)^2} + \sqrt{(x_1 - s_x^v)^2 + (x_2 - s_y^v)^2} - d^{(u,v)}. \tag{A.24}$$

The partial derivative of the function $\lambda^{(u,v)}(\mathbf{x})$ with respect to x_1 is given by

$$\begin{aligned}\frac{\partial}{\partial x_1}\lambda^{(u,v)}(\mathbf{x}) =& [(x_1 - s_x^u)^2 + (x_2 - s_y^u)^2]^{-1/2}(x_1 - s_x^u) + \\ & [(x_1 - s_x^v)^2 + (x_2 - s_y^v)^2]^{-1/2}(x_1 - s_x^v).\end{aligned} \tag{A.25}$$

Similarly, the partial derivative with respect to x_2 is given by

$$\begin{aligned}\frac{\partial}{\partial x_2}\lambda^{(u,v)}(\mathbf{x}) =& [(x_1 - s_x^u)^2 + (x_2 - s_y^u)^2]^{-1/2}(x_2 - s_y^u) + \\ & [(x_1 - s_x^v)^2 + (x_2 - s_y^v)^2]^{-1/2}(x_2 - s_y^v).\end{aligned} \tag{A.26}$$

Since the function $\lambda^{(u,v)}(\mathbf{x})$ does not involve the state variables x_3 and x_4, therefore the partial derivatives with respect to these variables are zero.

Appendix B

Distributed Filter Parameters for Different Communication Costs

In this section we present the filter parameters corresponding to performance results provided in Section 4.2.4. The communication cost of the distributed filtering schemes is dependent on the sample size and/or the number of gossip iterations. Hence, there are multiple filter settings that result in the same communication overhead. In the following tables we provide information about the sample size and the number of average gossip iterations that were used to generate the simulation results. In Table B.1, the filter parameters for the linear measurement model are presented which correspond to the performance results provided in Table 4.7. In some cases we were unable to find a working setting for a distributed filter at a specific cost, so we have marked it with a dash. The unscented Kalman filter does not have a sample size associated with it, therefore we inserted NA in its sample size field which means not applicable.

In Table B.2, the filter parameters for the range measurement model are presented which correspond to the performance results provided in Table 4.8.

In Table B.3, the filter parameters for the RF tomography measurement model are presented which correspond to the performance results provided in Table 4.9.

Table B.1 Distributed filter parameters (sample size and number of average gossip iterations) for the linear measurement model at different communication costs.

Scalars	8000		480		200	
	Sample size	Ave. gossip	Sample size	Ave. gossip	Sample size	Ave. gossip
GA [6]	2000	7000	1000	280	1000	100
LC [7]	1000	34000	1000	1120	1000	450
SM [8]	100	1000	12	430	12	140
Top-m [9]	500	700		-		-
UKF [78]	NA	5000	NA	250		-
UKF* [78]	NA	5000	NA	250	NA	75
EnKF	35	500	10	70		-
ESRF	100	5000	100	250	100	75
DEnKF	100	5000	100	250	100	75

Table B.2 Distributed filter parameters (sample size and number of average gossip iterations) for the range measurement model at different communication costs.

Scalars	16000		8000		800	
	Sample size	Ave. gossip	Sample size	Ave. gossip	Sample size	Ave. gossip
GA [6]	1000	16000	1000	7000	1000	425
LC [7]	1000	64000	1000	34000	1000	1800
SM [8]	200	1000	100	1000	25	300
Top-m [9]	800	850	500	700		-
UKF [78]	NA	10000	NA	4900	NA	350
EnKF	10	850	10	375		-
ESRF	12	800	10	436		-
DEnKF	12	800	10	436		-

Table B.3 Distributed filter parameters (sample size and number of average gossip iterations) for the RF tomography measurement model at different communication costs.

Scalars	16667		500		350	
	Sample size	Ave. gossip	Sample size	Ave. gossip	Sample size	Ave. gossip
GA [6]	1000	16000	1000	280	1000	200
LC [7]	1000	64000		-		-
SM [8]	200	1000		-		-
Top-m [9]	800	850		-		-
UKF [78]	NA	50000	NA	1500	NA	1000
EnKF		-		-		-
ESRF	100	50000	100	1450	100	1000
DEnKF	100	50000	100	1450	100	1000

References

[1] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 40–50, Mar. 2002.

[2] J. Polastre, R. Szewczyk, A. Mainwaring, D. Culler, and J. Anderson, "Analysis of wireless sensor networks for habitat monitoring," in *Wireless Sensor Netw.*, C. S. Raghavendra, K. M. Sivalingam, and T. Znati, Eds. Springer US, 2004, pp. 399–423.

[3] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Proc. IEEE Eur. Control Conf.*, Seville, Spain, Dec. 2007.

[4] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2069–2084, Sep. 2010.

[5] M. Coates, "Distributed particle filters for sensor networks," in *Proc. Int. Symp. Inf. Process. Sensor Netw. (IPSN)*, Berkeley, CA, USA, Apr. 2004.

[6] B. N. Oreshkin and M. J. Coates, "Asynchronous distributed particle filter via decentralized evaluation of Gaussian products," in *Proc. ISIF Int. Conf. Inf. Fusion*, Edinburgh, UK, July 2010.

[7] O. Hlinka, O. Sluciak, F. Hlawatsch, P. M. Djuric, and M. Rupp, "Likelihood consensus: Principles and application to distributed particle filtering," in *Proc. Asilomar Conf. Signals, Syst. Comp.*, Pacific Grove, CA, USA, Nov. 2010.

[8] S. Farahmand, S. I. Roumeliotis, and G. B. Giannakis, "Set-membership constrained particle filter: Distributed adaptation for sensor networks," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4122–4138, Sept. 2011.

[9] D. Üstebay, M. Coates, and M. Rabbat, "Distributed auxiliary particle filters using selective gossip," in *Proc. IEEE Int. Conf. Acoustics Speech and Signal Process. (ICASSP)*, Prague, Czech Republic, May 2011.

[10] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics," *J. Geophy. Res.-All Ser.*, vol. 99, pp. 10–10, May 1994.

[11] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[12] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, 1960.

[13] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *J. Basic Eng., ser. D*, vol. 83, pp. 95–108, Mar. 1961.

[14] Y. C. Ho and R. Lee, "A Bayesian approach to problems in stochastic estimation and control," *IEEE Trans. Autom. Control*, vol. 9, no. 4, pp. 333–339, 1964.

[15] G. Welch and G. Bishop, "An introduction to the Kalman filter," 1995.

[16] G. Evensen, "The ensemble Kalman filter: Theoretical formulation and practical implementation," *Ocean Dyn.*, vol. 53, no. 4, pp. 343–367, Nov. 2003.

[17] ——, *Data assimilation: the ensemble Kalman filter*. Berlin, Heidelberg: Springer-Verlag, 2009.

[18] G. Burgers, P. J. V. Leeuwen, and G. Evensen, "Analysis scheme in the ensemble Kalman filter," *Mon. Weather Rev.*, vol. 126, no. 6, pp. 1719–1724, 1998.

[19] A. Doucet, N. D. Freitas, and N. Gordon, "An introduction to sequential Monte Carlo methods," A. Doucet, N. D. Freitas, and N. Gordon, Eds. New York: Springer-Verlag, 2001, pp. 3–14.

[20] P. Sakov and P. R. Oke, "Implications of the form of the ensemble transformation in the ensemble square root filters," *Mon. Weather Rev.*, vol. 136, no. 3, pp. 1042–1053, Mar. 2008.

[21] ——, "A deterministic formulation of the ensemble Kalman filter: an alternative to ensemble square root filters," *Tellus A*, vol. 60, no. 2, pp. 361–371, Jan. 2008.

[22] G. Evensen and P. J. V. Leeuwen, "An ensemble Kalman smoother for nonlinear dynamics," *Mon. Weather Rev.*, vol. 128, no. 6, pp. 1852–1867, 2000.

[23] G. Evensen, "The ensemble Kalman filter for combined state and parameter estimation," *IEEE Control Syst. Mag.*, vol. 29, no. 3, pp. 83–104, Jun. 2009.

[24] J. D. Kepert, "On ensemble representation of the observation-error covariance in the ensemble Kalman filter," *Ocean Dyn.*, vol. 54, no. 6, pp. 561–569, 2004.

[25] C. H. Bishop, B. J. Etherton, and S. J. Majumdar, "Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects," *Mon. Weather Rev.*, vol. 129, no. 3, pp. 420–436, 2001.

[26] J. L. Anderson, "An ensemble adjustment Kalman filter for data assimilation," *Mon. Weather Rev.*, vol. 129, no. 12, pp. 2884–2903, 2001.

[27] J. S. Whitaker and T. M. Hamill, "Ensemble data assimilation without perturbed observations," *Mon. Weather Rev.*, vol. 130, no. 7, pp. 1913–1924, 2002.

[28] M. K. Tippett, J. L. Anderson, C. H. Bishop, T. M. Hamill, and J. S. Whitaker, "Ensemble square root filters*," *Mon. Weather Rev.*, vol. 131, no. 7, pp. 1485–1490, 2003.

[29] G. Evensen, "Sampling strategies and square root analysis schemes for the EnKF," *Ocean Dyn.*, vol. 54, no. 6, pp. 539–560, 2004.

[30] X. Wang, C. H. Bishop, and S. J. Julier, "Which is better, an ensemble of positive-negative pairs or a centered spherical simplex ensemble?" *Mon. Weather Rev.*, vol. 132, no. 7, pp. 1590–1605, 2004.

[31] O. Leeuwenburgh, G. Evensen, and L. Bertino, "The impact of ensemble filter definition on the assimilation of temperature profiles in the tropical pacific," *Q. J. R. Meteorol. Soc.*, vol. 131, no. 613, pp. 3291–3300, 2005.

[32] D. M. Livings, S. L. Dance, and N. K. Nichols, "Unbiased ensemble square root filters," *Physica D: Nonlinear Phenomena*, vol. 237, no. 8, pp. 1021–1028, 2008.

[33] S. Reich, "A non-parametric ensemble transform method for Bayesian inference," *arXiv preprint arXiv:1210.0375*, 2012.

[34] C. Villani, *Topics in optimal transportation*. Providence: American Math. Soc., 2003.

[35] G. Dantzig, *Linear programming and extensions*. Princeton University Press, 1998.

[36] C. J. Cotter and S. Reich, "Ensemble filter techniques for intermittent data assimilation-a survey," *arXiv preprint arXiv:1208.6572*, 2012.

[37] K. Bergemann, G. Gottwald, and S. Reich, "Ensemble propagation and continuous matrix factorization algorithms," *Q. J. R. Meteorol. Soc.*, vol. 135, no. 643, pp. 1560–1572, 2009.

[38] K. Bergemann and S. Reich, "A localization technique for ensemble Kalman filters," *Q. J. R. Meteorol. Soc.*, vol. 136, no. 648, pp. 701–707, 2010.

[39] J. Amezcua, K. Ide, E. Kalnay, and S. Reich, "Ensemble transform Kalman-Bucy filters," *Q. J. R. Meteorol. Soc.*, 2012.

[40] S. Aanonsen, G. Nævdal, D. Oliver, A. Reynolds, and B. Vallès, "The ensemble Kalman filter in reservoir engineering–a review," *J. Soc. Petroleum Eng.*, vol. 14, no. 3, pp. 393–412, 2009.

[41] L. Dovera and E. D. Rossa, "Multimodal ensemble Kalman filtering using Gaussian mixture models," *Comput. Geosci.*, vol. 15, no. 2, pp. 307–323, 2011.

[42] D. Alspach and H. W. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Trans. Autom. Control*, vol. 17, no. 4, pp. 439–448, 1972.

[43] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Stat. Soc., ser. B (Methodological)*, pp. 1–38, 1977.

[44] M. Frei and H. R. KüNsch, "Mixture ensemble Kalman filters," *Comput. Stat. Data Anal.*, vol. 58, pp. 127–138, Feb. 2013.

[45] 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.*, vol. 15, no. 2, pp. 293–305, 2011.

[46] T. Bengtsson, C. Snyder, and D. Nychka, "Toward a nonlinear ensemble filter for high-dimensional systems," *J. Geophy. Res.*, vol. 108, no. D24, p. 8775, 2003.

[47] J. H. Kotecha and P. M. Djuric, "Gaussian sum particle filtering," *IEEE Trans. Signal Process.*, vol. 51, no. 10, pp. 2602–2612, Oct. 2003.

[48] A. S. Stordal, R. Valestrand, H. A. Karlsen, G. Nævdal, and H. J. Skaug, "Comparing the adaptive Gaussian mixture filter with the ensemble Kalman filter on synthetic reservoir models," *Comput. Geosci.*, pp. 1–16, 2012.

[49] J. Rezaie and J. Eidsvik, "Shrinked (1- α) ensemble Kalman filter and α Gaussian mixture filter," *Comput. Geosci.*, pp. 1–16, 2012.

[50] S. Reich, "A Gaussian-mixture ensemble transform filter," *Q. J. R. Meteorol. Soc.*, vol. 138, no. 662, pp. 222–233, 2011.

[51] P. L. Houtekamer and H. L. Mitchell, "A sequential ensemble Kalman filter for atmospheric data assimilation," *Mon. Weather Rev.*, vol. 129, no. 1, pp. 123–137, 2001.

[52] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridgeshire, New York: Cambridge University Press, 1990.

[53] T. M. Hamill, J. S. Whitaker, and C. Snyder, "Distance-dependent filtering of background error covariance estimates in an ensemble Kalman filter," *Mon. Weather Rev.*, vol. 129, no. 11, pp. 2776–2790, 2001.

[54] G. Gaspari and S. E. Cohn, "Construction of correlation functions in two and three dimensions," *Q. J. R. Meteorol. Soc.*, vol. 125, no. 554, pp. 723–757, 1999.

[55] E. Ott, B. R. Hunt, I. Szunyogh, A. V. Zimin, E. J. Kostelich, M. Corazza, E. Kalnay, D. J. Patil, and J. A. Yorke, "A local ensemble Kalman filter for atmospheric data assimilation," *Tellus A*, vol. 56, no. 5, pp. 415–428, 2004.

[56] J. L. Anderson, "Exploring the need for localization in ensemble data assimilation using a hierarchical ensemble filter," *Physica D: Nonlinear Phenomena*, vol. 230, no. 1, pp. 99–111, 2007.

[57] E. J. Fertig, B. R. Hunt, E. Ott, and I. Szunyogh, "Assimilating non-local observations with a local ensemble Kalman filter," *Tellus A*, vol. 59, no. 5, pp. 719–730, 2007.

[58] C. H. Bishop and D. Hodyss, "Flow-adaptive moderation of spurious ensemble correlations and its use in ensemble-based data assimilation," *Q. J. R. Meteorol. Soc.*, vol. 133, no. 629, pp. 2029–2044, 2007.

[59] J. L. Anderson, "An adaptive covariance inflation error correction algorithm for ensemble filters," *Tellus A*, vol. 59, no. 2, 2007.

[60] H. Li, E. Kalnay, and T. Miyoshi, "Simultaneous estimation of covariance inflation and observation errors within an ensemble Kalman filter," *Q. J. R. Meteorol. Soc.*, vol. 135, no. 639, pp. 523–533, 2009.

[61] S. Reich, "Ensemble Kalman and H filters," in *AIP Conf. Proc.*, vol. 1281, no. 1, 2010, pp. 19–22.

[62] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. Wiley-Interscience, 2006.

[63] C. L. Keppenne, M. M. Rienecker, N. Kurkowski, and D. A. Adamec, "Ensemble Kalman filter assimilation of temperature and altimeter data with bias correction and application to seasonal prediction," *Nonlinear process. Geophysics*, vol. 12, no. 4, pp. 491–503, 2005.

[64] E. J. Fertig, S. J. Baek, B. R. Hunt, E. Ott, I. Szunyogh, J. A. Aravéquia, E. Kalnay, H. Li, and J. Liu, "Observation bias correction with an ensemble Kalman filter," *Tellus A*, vol. 61, no. 2, pp. 210–226, 2009.

[65] S. J. Baek, B. R. Hunt, E. Kalnay, E. Ott, and I. Szunyogh, "Local ensemble Kalman filtering in the presence of model bias," *Tellus A*, vol. 58, no. 3, pp. 293–306, 2006.

[66] G. A. Gottwald, L. Mitchell, and S. Reich, "Controlling overestimation of error covariance in ensemble Kalman filters with sparse observations: A variance limiting Kalman filter," *arXiv preprint arXiv:1108.5801*, 2011.

[67] R. J. Lorentzen and G. Nævdal, "An iterative ensemble Kalman filter," *IEEE Trans. Autom. Control*, vol. 56, no. 8, pp. 1990–1995, 2011.

[68] J. Mandel, L. Cobb, and J. D. Beezley, "On the convergence of the ensemble Kalman filter," *Applicat. Math.*, vol. 56, no. 6, pp. 533–541, 2011.

[69] R. Tavakoli, G. Pencheva, and M. Wheeler, "Multi-level parallelization of ensemble Kalman filter for reservoir history matching," in *Soc. Petroleum Eng., Reservoir Simul. Symp.*, 2011.

[70] T. Xu, J. J. G. Hernández, L. Li, and H. Zhou, "Parallelized ensemble Kalman filter for hydraulic conductivity characterization," *Comput. & Geosci.*, vol. 52, no. 0, pp. 42 – 49, 2013.

[71] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proc. Int. Symp. Inf. Process. Sensor Netw. (IPSN)*, Berkeley, CA, USA, Apr. 2004, pp. 20–27.

[72] D. Blatt and A. O. Hero, "Energy-based sensor network source localization via projection onto convex sets," *IEEE Trans. Signal Process.*, vol. 54, no. 9, pp. 3614–3619, 2006.

[73] S. Son, M. Chiang, S. R. Kulkarni, and S. C. Schwartz, "The value of clustering in distributed estimation for sensor networks," in *Proc. IEEE Int. Conf. Wireless Netw. Commun. Mobile Comput.*, vol. 2, 2005, pp. 969–974.

[74] Y. Yu, B. Krishnamachari, and V. K. Prasanna, "Energy-latency tradeoffs for data gathering in wireless sensor networks," in *Proc. IEEE INFOCOM 2004*, vol. 1, Hong Kong, Mar. 2004.

[75] A. Ciancio, S. Pattem, A. Ortega, and B. Krishnamachari, "Energy-efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm," in *Proc. IEEE/ACM Symp. Inf. Process. Sensor Netw. (IPSN)*, 2006, pp. 309–316.

[76] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu, "Data-centric storage in sensornets with ght, a geographic hash table," *Mobile Nets. Appl.*, vol. 8, no. 4, pp. 427–442, 2003.

[77] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proc. of the IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.

[78] W. Li and Y. Jia, "Consensus-based distributed multiple model UKF for jump Markov nonlinear systems," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 227–233, Jan. 2012.

[79] ——, "Distributed consensus filtering for discrete-time nonlinear systems with non-Gaussian noise," *Signal Process.*, vol. 92, no. 10, pp. 2464–2470, 2012.

[80] A. Simonetto, T. Keviczky, and R. Babuska, "Distributed nonlinear estimation for robot localization using weighted consensus," in *Proc. IEEE Int. Control Robot. Autom. (ICRA)*, Anchorage, AK, USA, May 2010, pp. 3026–3031.

[81] A. Mohammadi and A. Asif, "Consensus-based distributed unscented particle filter," in *Proc. IEEE Stat. Signal Process. Workshop (SSP)*, Nice, France, Jun. 2011.

[82] ——, "A constraint sufficient statistics based distributed particle filter for bearing only tracking," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 3670–3675.

[83] D. Üstebay, R. Castro, M. Coates, and M. Rabbat, "Distributed approximation and tracking using selective gossip," in *Compressed Sensing and Sparse Filtering*, A. Y. Carmi, L. S. Mihaylova, and S. J. Godsill, Eds. Berlin: Springer-Verlag, 2014 (to be published).

[84] J. Elson and K. Römer, "Wireless sensor networks: A new regime for time synchronization," vol. 33, no. 1, New York, NY, USA, Jan. 2003, pp. 149–154.

[85] B. D. O. Anderson and J. B. Moore, *Optimal filtering.* Englewood Cliffs: Prentice-Hall, 1979.

[86] J. Speyer, "Computation and transmission requirements for a decentralized linear-quadratic-gaussian control problem," *IEEE Trans. Autom. Control*, vol. 24, no. 2, pp. 266–269, 1979.

[87] MATLAB, *version 8.1.0 (R2013a).* Natick, Massachusetts: The MathWorks Inc., 2013.

[88] B. Ristic, S. Arulampalm, and N. J. Gordon, *Beyond the Kalman filter: Particle filters for tracking applications.* Artech House Publishers, 2004.

[89] X. Chen, A. Edelstein, Y. Li, M. J. Coates, M. G. Rabbat, and A. Men, "Sequential monte carlo for simultaneous passive device-free tracking and sensor localization using received signal strength measurements," in *Proc. IEEE Symp. Inf. Process. Sensor Netw. (IPSN)*, Chicago, IL, USA, Apr. 2011, pp. 342–353.

[90] F. Thouin, S. Nannuru, and M. J. Coates, "Multi-target tracking for measurement models with additive contributions," in *Int. Conf. Inf. Fusion*, Chicago, IL, USA, Jul. 2011, pp. 1–8.

[91] D. Lee, "Nonlinear estimation and multiple sensor fusion using unscented information filtering," *IEEE Signal Process. Lett.*, vol. 15, pp. 861–864, 2008.