NONLINEAR FILTERING WITH VARIOUS IRREGULARITIES IN MEASUREMENT DATA

Ph.D. Thesis

By

GUDDU KUMAR

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY INDORE
December 2022

Nonlinear Filtering with Various Irregularities in Measurement Data

A THESIS

Submitted in partial fulfillment of the

requirements for the award of the degree

of

DOCTOR OF PHILOSOPHY

by

GUDDU KUMAR

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

December 2022

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled "**Nonlinear Filtering with Various Irregularities in Measurement Data**" in the partial fulfillment of the requirements for the award of the degree of DOCTOR OF PHILOSOPHY and submitted in the DEPARTMENT OF ELECTRICAL ENGINEERING, Indian Institute of Technology Indore, is an authentic record of my own work carried out during the time period from June 2019 to December 2022 under the supervision of Dr. Abhinoy Kumar Singh and Dr. Swaminathan R, Indian Institute of Technology Indore, India.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

Guddu Kumar

Signature of the student with date

(GUDDU KUMAR)

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

09/05/23

Signature of Thesis Supervisor with date

(Dr. Abhinoy Kumar Singh)

09/05/2023

Signature of Thesis Supervisor with date

(Dr. Swaminathan R)

Guddu Kumar has successfully given his Ph.D. Oral Examination held on May, 08, 2023.

ACKNOWLEDGEMENTS

My life has been filled with some of the best moments in the last four years. Over the course of this period, I have received support and encouragement from many people. The completion of this thesis would not have been possible without the patience and support of these individuals. I would like to extend my sincere gratitude to them. Firstly, I would like to express my deepest gratitude to my supervisors, Dr. Abhinoy Kumar Singh and Dr. Swaminthan R, for their excellent guidance, constant support, and positive encouragement over the years. I appreciate their detailed, prompt, and insightful feedback on all manuscripts that I sent to them at different stages of my Ph.D. work. I am truly thankful to them for spending their valuable time on my thesis works. From my supervisors, I didn't learn only about the technical aspects of my research, but I also learn about how to be optimistic during the research progress. My appreciation goes out to their patient mentoring and the wealth of knowledge they have imparted: not just in how to solve challenging problems but in writing and presenting papers clearly as well.

Furthermore, I am grateful to Prof. Vimal Bhatia and Prof. Ram Bilas Pachori, for their insightful, thoughtful, and detailed comments on my different research works. I am also thankful to Dr. Paresh Date. I learn from him an enthusiastic attitude and how to consider problems from a border perspective in research. I am also grateful to PSPC members, Prof. Prabhat Kumar Upadhyay and Dr. Neminath Hubballi, for their insightful guidance during the research advancement.

I would like to extend my heartfelt gratitude to my lab mates, Mr. Sumanta Kuman Nanda, Mr. Amit Kumar Naik, and Mr. Yamalakonda Venu Gopal, for their unwavering support throughout my Ph.D. journey.

Finally, I dedicate this thesis to my family, particularly my parents, for their constant and unconditional love and support throughout my life. I am also deeply grateful to my partner, Ankita Kumari, for her unwavering support and selfless sacrifices. Without her love and understanding, completing this thesis would have been significantly more challenging.

GUDDU KUMAR

Dedicated

to
My family

ABSTRACT

Estimation is a popular computational tool for determining the internal states of a dynamical system from noisy measurements. A recursive process of estimation is called filtering. The conceptual filtering solution is obtained in terms of unknown probability density functions (PDF). Several analytical filtering solutions have been presented in the literature by characterizing the unknown PDFs differently. The popularly known Kalman filter is an optimal analytical filter for linear dynamical systems, while an optimal nonlinear filter is still a future scope. This thesis is particularly concerned with suboptimal nonlinear filtering.

There are two popular nonlinear filtering methods, namely Gaussian filtering and particle filtering. The Gaussian filtering approximates the unknown PDFs as well as the unknown noises as Gaussian. The particle filtering characterizes the unknown PDFs as a weighted summation of particles. The literature beholds many variants of the Gaussian filtering as well as the particle filtering, availing an impressive trade-off between accuracy and computational demand. Therefore, if an adequate computational budget is available, under the general problem scenarios, the accuracy may not be a serious concern despite the suboptimality of nonlinear filtering. Although the practical problems often perceive complicated scenarios, where the existing nonlinear filters fail or underperform.

This thesis is concerned with various complicated scenarios, which are defined in terms of different measurement irregularities. For example, the transmission and processing of measurement data through cyber-physical systems introduce the risks of delay, intermittently missing, and cyber-attacks. Also, noises that inherently appear in measurements can be non-Gaussian, and a Gaussian approximation (in Gaussian filtering) can often considerably deteriorate the accuracy. Furthermore, the measurement data (as well as the unknown states) consisting of angular information is of finite-range, while the traditional nonlinear filters are designed for infinite-range data, which causes poor accuracy for range-limited data like angular data.

Summarizing the above discussions, this thesis aims to address various measurement irregularities, including the delay, intermittently missing measurements, cyber-attacks, non-Gaussian measurement noises, and angular data. In this regard, this thesis contributes with the following modifications in the existing Gaussian filtering and particle filtering.

- The thesis introduces an advanced Gaussian filtering method for handling large delays with a reduced number of prerequisite probabilities. It also precludes the need of assigning any

upper bound of delay, which is an essential requirement in the existing extensions of Gaussian filtering for handling the delayed measurements. Ultimately, this method improves the filtering accuracy.

- In another contribution, this thesis develops an advanced Gaussian filtering algorithm for handling delayed measurements under a non-Gaussian noisy environment. Interestingly, this contribution is applicable for fractional delays (delay being a fraction of the sampling interval) unlike the existing methods that can handle only integer delay.

- This thesis develops a new Gaussian filtering method for handling cyber-attacked measurements. Interestingly, the developed method can handle different forms of the data-tampering (due to the cyber-attack), including false data injection, de-synchronizing of data, and denial-of-services.

- The thesis redesigns the traditional particle filtering method for efficiently handling the finite-range angular data. In this regard, probably for the first time, the proposed method introduces wrapped normal distribution in particle filtering.

Contents

List of Figures

List of Tables

List of Abbreviations

ARMSE	Average Root Mean Square Error
CIM	Correntropy Induced Metric
CKF	Cubature Kalman Filter
CF	Circular Filter
CPF	Cubature Particle Filter
DoS	Denial-of-Service
EKF	Extended Kalman Filter
FDI	False Data Injection
GHF	Gaussian Hermite Filter
GHPF	Gaussian Hermite Particle Filter
MMSE	Minimum Mean Square Error
MC	Maximum Correntropy
MCCKF	Maximum Correntropy with Cubature Kalman Filter
MLCKF	Maximum Likelihood based Cubature Kalman Filter
PDF	Probability Density Function
PF	Particle Filter
RD	Random Delay
RSPF	Rogers-Szegő Particle Filter
RMSE	Root Mean Square Error
TAM	Time Asynchronous Measurement
UKF	Unscented Kalman Filter
UPF	Unscented Particle Filter
WN	Wrapped Normal

List of Symbols

$\mathbf{x}_k$	State variable
θ_k	Angular state variable
$\mathbf{z}_k$	Measurement variable
$\mathbf{y}_k$	Measurement variable under different irregularities
n	Dimension of state variable
r	Dimension of measurement variable
$\mathbb{R}^n$	n-dimensional column vector of all real numbers
t_k	k^{th} sampling instant
τ	Sampling interval
T_s	Number of time steps
T_{mc}	Number of Monte-Carlo runs
ν_k	Process noise
η_k	Measurement noise
$\mathbf{Q}_k$	Covariance of process noise
$\mathbf{R}_k$	Covariance of measurement noise
$\phi(\cdot)$	General nonlinear function of process model
$\Psi(\cdot)$	General nonlinear function of measurement model
$\mathbb{F}(\cdot)$	General nonlinear function
$\mathbb{E}[\cdot]$	Statistical expectation operator
$\mathcal{N}(\cdot)$	Gaussian distribution
$\boldsymbol{\Upsilon}$	Set of deterministic sample points
$\boldsymbol{W}$	Set of deterministic weights associated with sample points
N_s	Number of deterministic sample points

N_p	Number of stochastic sample points or particles
$\mathbf{I}_n$	n-dimensional identity matrix
$chol(\cdot)$	Cholesky decomposition
$\hat{\mathbf{x}}_{k\|k-1}$	Predicted state
$\mathbf{P}_{k\|k-1}$	Predicted error covariance
$\hat{\mathbf{z}}_{k\|k-1}$	Predicted measurement
$\hat{\mathbf{y}}_{k\|k-1}$	Predicted measurement under different irregularities
$\mathbf{P}^{\mathbf{zz}}_{k\|k-1}$	Measurement error covariance
$\mathbf{P}^{\mathbf{yy}}_{k\|k-1}$	Measurement error covariance under different irregularities
$\mathbf{P}^{\mathbf{xz}}_{k\|k-1}$	Cross-covariance of state and measurement
$\mathbf{P}^{\mathbf{xy}}_{k\|k-1}$	Cross-covariance of state and measurement under different irregularities
α_k, β_k,γ_k	Bernoulli random variables
$G_{d,k}$	Geometric random variable for modeling d-delay at time t_k
M_d	Maximum number of delays
M_s	Maximum number of intermediate instants between two sampling instants
δ	Intermediate sampling interval
t_g^d	An intermediate instant
t_g	Set of past sampling instants
N_t	Set of past intermediate instants
$\mathbb{L}_{g,d}$	Gaussian log-likelihood at t_g^d
$t_{\tilde{g}}^{\tilde{d}}$	A past intermediate instant with maximum likelihood of arrival of the delayed measurement $\mathbf{y}_k$
$\hat{\mathbf{x}}_{g,d\|k-1}$	Predicted state at t_g^d
$\mathbf{P}_{g,d\|k-1}$	Predicted error covariance at t_g^d
$\hat{\mathbf{z}}_{g,d\|k-1}$	Predicted measurement at t_g^d
$\hat{\mathbf{x}}_{\tilde{g},\tilde{d}\|k}$	Estimated state at $t_{\tilde{g}}^{\tilde{d}}$
$\mathbf{P}_{\tilde{g},\tilde{d}\|k}$	Estimated error covariance at $t_{\tilde{g}}^{\tilde{d}}$
$\mathbf{K}_k$	Kalman gain
$\hat{\mathbf{x}}_{k\|k}$	Estimated state
$\mathbf{P}_{k\|k}$	Estimated error covariance

Chapter 1

Introduction

1.1 Background

Modern engineering and non-engineering applications [1] widely utilize data. The data may be characterized into various types, depending upon its source. For example, sensor-generated data, experimental data, and survey data. Regardless, irrespective of its source, the data commonly involve errors, technically known as noises. Therefore, it requires potential computational methods [2] for analyzing and processing the noisy data.

Estimation is a typical computational tool for determining the internal states of a dynamical system from the received noisy data, see Fig.1.1. A recursive process of estimation is popularly known as filtering [3].

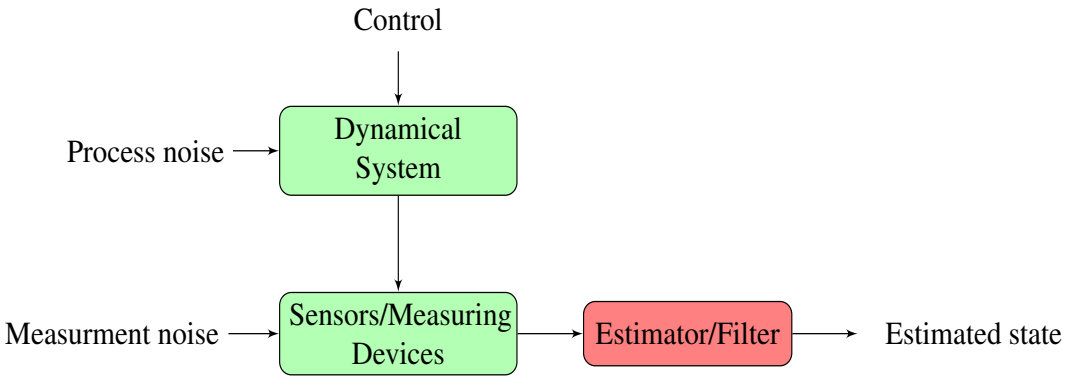


Figure 1.1: Schematic representation of filtering

In the estimation and filtering theory [4], the received data is commonly known as measurements. The estimation and filtering find widespread applications in engineering as well as

non-engineering domains. Some of the popular domains of applications can be mentioned as target tracking [3], network control and communication systems [5], space technology [6], fault diagnosis [7], biomedical system [8], robotics [9], industrial diagnosis and prognosis [10], navigation [11], financial modeling and monitoring [12], weather forecasting [13], pandemic monitoring [14], *etc*. Hereafter, the author will simply refer to the term 'filtering' to demonstrate the term 'filtering' as well as 'estimation'.

The filtering is a model-based computational tool [15], requiring the state space model of dynamical systems. The state space model consists of the process and measurement models, with the following descriptions:

- Process model: It characterizes the dynamical behavior of the states. Specifically, it consists of noises to compensate for the modeling errors of the true dynamics.

- Measurement model: It characterizes the mathematical relationship between the observed data and the unknown states. It is additionally having a noise component to compensate for the observation errors, which may be due to device errors and data processing errors.

Considering the above descriptions, the general forms of the process and measurement models are given below.

Process model

$$\mathbf{x}_k = \Phi_{k-1}(\mathbf{x}_{k-1}) + \nu_{k-1}. \tag{1.1}$$

Measurement model

$$\mathbf{z}_k = \Psi_k(\mathbf{x}_k) + \eta_k, \tag{1.2}$$

where $\mathbf{x}_k \in \mathbb{R}^n$ and $\mathbf{z}_k \in \mathbb{R}^r$ are the state and measurement variables, respectively $\forall k \in \{1,2,\cdots\}$. Herewith, $\Phi_k : \mathbf{x}_{k-1} \to \mathbf{x}_k$ and $\Psi_k : \mathbf{x}_k \to \mathbf{z}_k$ denote general mathematical functions, while ν_k and η_k represent the process and measurement noises, respectively. In general, the process and measurement noises are assumed to be additive, white, and Gaussian [16]. This thesis also follows these assumptions unless they are not dismissed through explicit statements.

To this end, the author defines the filtering objective as a recursive process of estimating the unknown states $\mathbf{x}_k$ $\forall k \in \{1,2,\cdots\}$, as the measurement $\mathbf{z}_k$ is sequentially received $\forall k \in \{1,2,\cdots\}$. The popularly known Kalman filter [17, 18] is an optimal filtering method, but it is limited to linear dynamical systems only. Therefore, this thesis is explicitly concerned with nonlinear filtering,

requiring either or both of $\Phi_k : \mathbf{x}_{k-1} \rightarrow \mathbf{x}_k$ and $\Psi_k : \mathbf{x}_k \rightarrow \mathbf{z}_k$ to be nonlinear.

The literature consists of two popular nonlinear filtering approaches, namely the Gaussian filtering [19] and particle filtering [20]. In the upcoming sections, we first introduce the Gaussian filtering and then discuss about the particle filtering.

1.2 Gaussian filtering

The Gaussian filtering is performed under the popularly known Bayesian filtering framework. Therefore, this section first introduces the Bayesian filtering framework. Henceforth, the discussion on simplification and formulation of the Bayesian filtering framework into the Gaussian filtering is provided.

1.2.1 Bayesian filtering framework

Here, the author highlights the Bayesian filtering framework [21], giving the desired estimate of state $\mathbf{x}_k$ in terms of the probability density function (PDF) $P(\mathbf{x}_k|\mathbf{z}_{1:k})$. As it gives the filtering solution in terms of PDF, it does not provide an analytical solution [22]. In particular, it principally formulates the core filtering theory for designing the Gaussian filters.

The Bayesian filtering obtains the desired PDF $P(\mathbf{x}_k|\mathbf{z}_{1:k})$ into two steps, namely the prediction and update steps, which are also known as time update and measurement update steps, respectively [23]. The prediction step obtains the prior PDF $P(\mathbf{x}_k|\mathbf{z}_{1:k-1})$, while the update step determines the posterior PDF $P(\mathbf{x}_k|\mathbf{z}_{1:k})$. In the following discussions, the author illustrates the theoretical and mathematical aspects of the prediction and update steps.

Prediction:

The prediction step determines the prior PDF $P(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ to predict the desired state one-step forward in time. In this regard, it utilizes the popularly known Chapman-Kolmogorov [2] equation, giving

$$P(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int_{\mathbb{R}^n} P(\mathbf{x}_k|\mathbf{x}_{k-1})P(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1}. \tag{1.3}$$

Update:

The update step obtains the desired posterior PDF $P(\mathbf{x}_k|\mathbf{z}_{1:k})$ by correcting the predicted PDF $P(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ using the information received from the latest measurement $\mathbf{z}_k$, arriving at t_k. In this regard, it applies the popularly known Baye's rule, which gives

$$P(\mathbf{x}_k|\mathbf{z}_{1:k}) = P(\mathbf{x}_k|\mathbf{z}_{1:k-1},\mathbf{z}_k,) = \frac{1}{c_k} P(\mathbf{z}_k|\mathbf{x}_k) P(\mathbf{x}_k|\mathbf{z}_{1:k-1}), \tag{1.4}$$

where $P(\mathbf{z}_k|\mathbf{x}_k)$ is the measurement likelihood function and c_k is a normalization constant, given as

$$c_k = P(\mathbf{z}_k|\mathbf{z}_{1:k-1}) = \int_{\mathbb{R}^n} P(\mathbf{z}_k|\mathbf{x}_k) P(\mathbf{x}_k|\mathbf{z}_{1:k-1}) d\mathbf{x}_k.$$

It is conclusive from Eqs. (1.3) and (1.4) that the Bayesian filtering framework provides a probabilistic solution, which is inadequate for accomplishing an analytical estimate of $\mathbf{x}_k$. Eventually, a schematic representation of the Bayesian filter is shown in Fig. 1.2. As discussed earlier, Gaussian filtering is a popular simplification of the Bayesian framework for obtaining an analytical solution.

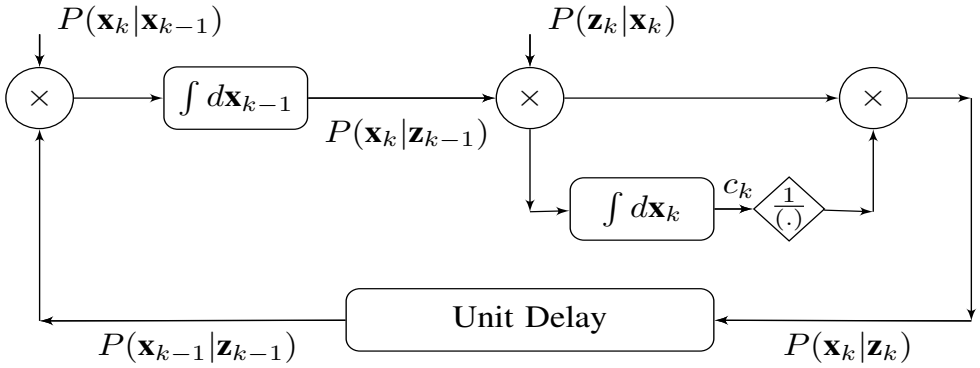


Figure 1.2: Schematic representation of recursive Bayesian filter in each recursion.

Hereafter, for better readability, the author considers the following notational simplicity: $P(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ is denoted as $P(\mathbf{x}_{k|k-1})$, while $P(\mathbf{x}_k|\mathbf{z}_{1:k})$ is denoted as $P(\mathbf{x}_{k|k})$. The author briefly discusses Gaussian filtering in the following section.

1.2.2 Simplification of Bayesian filtering as Gaussian filtering

The Gaussian filtering, as a simplification of the Bayesian filtering framework [24], is based on Gaussian approximations of various PDFs and noises, as discussed below.

- The Gaussian filtering approximates the various conditional PDFs that appeared in the Bayesian filtering as Gaussian, *i.e.*,

$$P(\mathbf{x}_{k|k-1}) \approx \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}), \tag{1.5}$$

$$P(\mathbf{x}_{k|k}) \approx \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}), \tag{1.6}$$

  and

$$P(\mathbf{z}_{k|k-1}) \approx \mathcal{N}(\mathbf{z}_k; \hat{\mathbf{z}}_{k|k-1}, \mathbf{P}^{\mathbf{zz}}_{k|k-1}), \tag{1.7}$$

  where $\mathcal{N}(\cdot)$ represents the Gaussian distribution, whereas $\hat{\mathbf{x}}_{k|k-1}$, $\hat{\mathbf{x}}_{k|k}$, $\mathbf{P}_{k|k-1}$, and $\mathbf{P}_{k|k}$ denote the prior estimate, posterior estimate, prior covariance, and posterior covariance of $\mathbf{x}_k$, respectively, while $\hat{\mathbf{z}}_{k|k-1}$ and $\mathbf{P}^{\mathbf{zz}}_{k|k-1}$ denote the predicted estimate and covariance of $\mathbf{z}_k$, respectively.

- The noises v_k and η_k are assumed as uncorrelated and approximated as zero-mean Gaussian with covariances $\mathbf{Q}_k$ and $\mathbf{R}_k$, respectively. Thus, we get $\mathbb{E}[\eta_k] = \mathbb{E}[\mathbf{v}_k] = \mathbb{E}[\eta_k \mathbf{v}_k^T] = 0$, $\eta_k \sim \mathcal{N}(0, \mathbf{R}_k)$ and $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{Q}_k)$, where $\mathbb{E}[\cdot]$ denotes the statistical expectation operator, while $\mathbf{R}_k = \mathbb{E}[\eta_k \eta_k^T]$ and $\mathbf{Q}_k = \mathbb{E}[\mathbf{v}_k \mathbf{v}_k^T]$.

With the above approximations, as followed from [25],[23], the computational aspects of the prediction and update steps for the Gaussian filtering are discussed below.

Prediction:

The Gaussian filtering computes the prior PDF using $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$, as [25],[23]

$$\hat{\mathbf{x}}_{k|k-1} \approx \int_{\mathbb{R}^n} \phi_{k-1}(\mathbf{x}_{k-1}) \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}) d\mathbf{x}_{k-1}, \tag{1.8}$$

$$\begin{aligned} \mathbf{P}_{k|k-1} \approx & \int_{\mathbb{R}^n} \phi_{k-1}(\mathbf{x}_{k-1}) \phi_{k-1}(\mathbf{x}_{k-1})^T \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1} \mathbf{P}_{k-1|k-1}) d\mathbf{x}_{k-1} \\ & - (\hat{\mathbf{x}}_{k-1|k-1})(\hat{\mathbf{x}}_{k-1|k-1})^T + \mathbf{Q}_k. \end{aligned} \tag{1.9}$$

Update:

This step computes the posterior estimate and covariance [25],[23], $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$, respectively. This extent, it updates the prior estimate and covariance, $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$, respectively using the noisy information of $\mathbf{z}_k$. To this regard, the computation of $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ ultimately requires the estimate and covariance of measurement, $\hat{\mathbf{z}}_{k|k-1}$ and $\mathbf{P}^{\mathbf{zz}}_{k|k-1}$, as well as the cross-covariance between the state and measurement, $\mathbf{P}^{\mathbf{xz}}_{k|k-1}$. Therefore, before determining the desired $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$, the author computes $\hat{\mathbf{z}}_{k|k-1}$, $\mathbf{P}^{\mathbf{zz}}_{k|k-1}$, and $\mathbf{P}^{\mathbf{xz}}_{k|k-1}$, respectively, as

$$\hat{\mathbf{z}}_{k|k-1} = \mathbb{E}[(\Psi_k(\mathbf{x}_k) + \mathbf{v}_k)\,|\mathbf{z}_{1:k-1}] \approx \int_{\mathbb{R}^n} \Psi_k(\mathbf{x}_k)\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})d\mathbf{x}_k, \tag{1.10}$$

$$\begin{aligned}\mathbf{P}^{\mathbf{zz}}_{k|k-1} &= \mathbb{E}[(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})^T] \\ &\approx \int_{\mathbb{R}^n} \Psi_k(\mathbf{x}_k)\Psi_k(\mathbf{x}_k)^T\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})d\mathbf{x}_k - (\hat{\mathbf{z}}_{k-1|k-1})(\hat{\mathbf{z}}_{k-1|k-1})^T + \mathbf{R}_k,\end{aligned} \tag{1.11}$$

$$\begin{aligned}\mathbf{P}^{\mathbf{xz}}_{k|k-1} &= \mathbb{E}[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})^T] \\ &\approx \int_{\mathbb{R}^n} \mathbf{x}_k\Psi_k(\mathbf{x}_k)^T\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})d\mathbf{x}_k - (\hat{\mathbf{x}}_{k|k-1})(\hat{\mathbf{z}}_{k|k-1})^T.\end{aligned} \tag{1.12}$$

Specifically, $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ are determined as

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}), \tag{1.13}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{P}^{\mathbf{zz}}_{k|k-1}\mathbf{K}_k^T, \tag{1.14}$$

where $\mathbf{K}_k$ denotes the Kalman gain [15], given as

$$\mathbf{K}_k = \mathbf{P}^{\mathbf{xz}}_{k|k-1}(\mathbf{P}^{\mathbf{zz}}_{k|k-1})^{-1}. \tag{1.15}$$

To this end, the Gaussian filtering [22] can be implemented through Eqs. (1.10) to (1.15). However, Eqs. (1.10) to (1.12) involve Gaussian weighted integrals of the form

$$I(\mathbb{F}) = \int_{\mathbb{R}^n} \mathbb{F}(\mathbf{x})\mathcal{N}(\mathbb{F}; \hat{\mathbf{x}}, \mathbf{P})d\mathbf{x}, \tag{1.16}$$

where $\mathbf{x}$ is a random variable with mean $\hat{\mathbf{x}}$ and covariance $\mathbf{P}$, and $\mathbb{F} : \mathbb{R}^n \to \mathbb{R}^n$ is a simple function.

The numerical methods used for integral approximation are generally defined for standard Gaussian, *i.e.*, for $\mathscr{N}(\mathbf{x};\mathbf{0}_{n\times 1},\mathbf{I}_n)$ with $\mathbf{I}_n$ be an n-dimensional unit matrix and $\mathbf{0}_{n\times 1}$ be an n-dimensional array of all zero elements. The author denotes this integral as $I_0(\mathbb{F})$, *i.e.*,

$$I_0(\mathbb{F}) = \int_{\mathbb{R}^n} \mathbb{F}(\mathbf{x})\mathscr{N}(\mathbf{x};\mathbf{0}_{n\times 1},\mathbf{I}_n)d\mathbf{x}. \tag{1.17}$$

The analytical solutions of such integrals exist for linear systems, having linear $\Phi_k : \mathbf{x}_{k-1} \rightarrow \mathbf{x}_k$ and $\Psi_k : \mathbf{x}_k \rightarrow \mathbf{z}_k$. Such a solution is popularly known as the Kalman filter [17]. Nonetheless, in the case of nonlinear systems, $\Phi_k : \mathbf{x}_{k-1} \rightarrow \mathbf{x}_k$ and $\Psi_k : \mathbf{x}_k \rightarrow \mathbf{z}_k$ (either or both) are nonlinear. Ultimately, the desired integral (Eq. (1.17)) appear in the form of '*nonlinear function* $\times$ *Gaussian distribution*', which is mostly intractable [19]. Consequently, the Gaussian filtering fails to accomplish an analytical solution. To get an analytical solution, two approaches are popular, which can be referred to as the derivative-based Gaussian filtering [16] and derivative-free Gaussian filtering [25].

In the derivative-based Gaussian filtering, the nonlinear dynamics are locally linearized using the derivatives. Thereafter, conclusively the linear Kalman filtering-based approach is applied. The readers may refer to [16, 26] for a detailed filtering algorithm for the derivative-based Gaussian filtering, popularly known as extended Kalman filtering. On the other hand, the derivative-free Gaussian filters utilize numerical methods for approximating $I_0(\mathbb{F})$ with the help of deterministically chosen sets of sample points and weights. Let us denote the sets of sample points and weights, respectively as $\boldsymbol{\Upsilon}$ and $\mathbf{W}$, respectively. Then, $I_0(\mathbb{F})$ is approximated as

$$I_0(\mathbb{F}) \approx \sum_{i=1}^{N_s} W_i \mathbb{F}(\boldsymbol{\Upsilon}_i), \tag{1.18}$$

where N_s is the number of sample points, and $\boldsymbol{\Upsilon}_i$ and W_i $\forall i \in \{1,2,\cdots,N_s\}$ are the i^{th} sample point and weight, respectively. The same numerical method can be extended for $\mathscr{N}(\mathbf{x};\hat{\mathbf{x}},\mathbf{P})$ by transforming $\boldsymbol{\Upsilon}$ with mean $\hat{\mathbf{x}}$ and covariance $\mathbf{P}$. Consequently, the desired intractable integral $I(\mathbb{F})$ is approximated as

$$I(\mathbb{F}) \approx \sum_{i=1}^{N_s} W_i \mathbb{F}(\hat{\mathbf{x}} + \boldsymbol{\Sigma}\boldsymbol{\Upsilon}_i), \tag{1.19}$$

where $\boldsymbol{\Sigma}\boldsymbol{\Sigma}^T = \mathbf{P}$. The author refers to [19],[23],[25] for a detailed filtering algorithm of the derivative-free Gaussian filtering.

1.3 Particle filtering

The particle filtering [20] realizes the desired posterior density using stochastically generated sample points, also known as particles, and their associated weights. Interestingly, it precludes Gaussian approximation of arbitrary PDFs as well as the presence of intractable integrals. In particular, it outperforms the above-discussed Gaussian filtering. As a result, particle filtering utilizes a large set of particles and weights for characterizing the true PDFs. Eventually, its computational demand is substantially larger than the Gaussian filtering. As a matter of fact, the particle filtering without re-sampling is essentially a Monte Carlo integration method [19]. In the past, particle filtering was mostly ignored in practical applications due to its high computational demand. Conversely, the recent years have exhibited substantial developments in efficient and cost-effective computational devices. Apparently, the particle filtering applications have been on the rise in developing high-precision tools and technologies.

Following the previous discussions, the estimation of $\mathbf{x}_k$ from $\mathbf{z}_k$ requires characterizing the PDF $P(\mathbf{x}_k|\mathbf{z}_k)$ analytically. In this regard, the particle filter approximates $P(\mathbf{x}_k|\mathbf{z}_k)$ as a weighted summation of stochastically generated sample points. The particle should be ideally generated from $P(\mathbf{x}_k|\mathbf{z}_k)$, which is unknown. Therefore, the particle filtering introduces a representative PDF $q(\mathbf{x}_k|\mathbf{z}_k)$, which may be conveniently obtained. The representative PDF $q(\mathbf{x}_k|\mathbf{z}_k)$ is broadly known as proposal density. Conversely, the particles are sampled from proposal density $q(\mathbf{x}_k|\mathbf{z}_k)$ to characterize the unknown PDF $P(\mathbf{x}_k|\mathbf{z}_k)$. The author denotes $\mathbf{x}_k^i$ and ω_k^i, $\forall i \in \{1,2,\cdots,N_p\}$, as the i^{th} particle and the associated weight, respectively, at t_k instant. The weights are normalized, *i.e.*, $\sum_{i=1}^{N_p} \omega_k^i = 1$. Subsequently, $P(\mathbf{x}_k|\mathbf{z}_k)$ is approximated as

$$P(\mathbf{x}_k|\mathbf{z}_k) \approx \sum_{i=1}^{N_p} \omega_k^i \delta\left(\mathbf{x}_k - \mathbf{x}_k^i\right), \tag{1.20}$$

where $\delta(\cdot)$ represents dirac delta function and N_p represents the number of particles [20].

1.4 Motivation

As mentioned previously, this thesis is primarily concerned with nonlinear filtering. In the nonlinear filtering, the literature still awaits for a closed-form solution, although two filtering approaches, *viz.* the Gaussian filtering and particle filtering gained significant popularity. Thankfully,

even though the closed-form solution is missing, several variants of Gaussian filtering and particle filtering are available for filtering under different system environments, precision demand, and computational budget allocation. For example, in a wide sense, we can state the following: i) the computationally efficient Gaussian filtering can be used if the computational budget allocation is small but the precision demand is not high and ii) the particle filtering may be utilized to accomplish high precision if the allocated computational budget is high.

In a serious concern, the nonlinear filtering, in general, ignores various measurement data irregularities, causing poor accuracy or even estimation failure [22]. Some of the popular measurement data irregularities are random delay in measurement, intermittently missing measurements, cyber-attacked measurements, range-limited data, intrinsic non-Gaussian measurement noises, *etc*. This thesis generally focuses on nonlinear filtering under various measurement data irregularities. In this regard, the particular motivations of this thesis are summarized below.

- The traditional Gaussian filtering straightforwardly ignores time-delays in measurements, while the practical measurement data often witness unknown and time-varying delays. The preliminary developments for handling this problem essentially require information on a large number of delay probabilities, which is difficult to characterize precisely. This thesis motivation is to develop advanced Gaussian filtering algorithms with the reduced number of probability requirements in handling the delayed measurements.

- The preliminary filtering algorithms that are designed for handling the different measurement irregularities mostly handle only one irregularity at a time. However, practically, the simultaneous occurrences of multiple irregularities can not be denied. This thesis is motivated to develop advanced Gaussian filtering algorithms for handling simultaneous occurrences of multiple measurement irregularities.

- The cyber-attack is becoming a challenging problem in the filtering applications involving data transmission and processing through cyber-physical systems. The motivation of this thesis is to design an advanced Gaussian filtering algorithm, which is robust to cyber-attacks.

- The traditional nonlinear filtering algorithms as well as their general extensions are designed for infinite-range data, while a practical data may often be of finite-range. For example, the angular data can vary in the range of $(-\pi, \pi]$, while the general nonlinear

filtering algorithms treat it over the entire real number. This thesis is motivated to develop a nonlinear filtering algorithm that treats the angular data within $(-\pi, \pi]$ to improve the accuracy.

It is important to note that the process model can be adjusted based on practitioner hypotheses to improve modeling, while the measurement model cannot be similarly altered due to its reliance on sensor-induced phenomena. Thus, in evaluating filter performance, an L2 norm metric is appropriate because, in this thesis, it assumes that true states are affected only by Gaussian noise and measurement noise potentially corrupted by both Gaussian and non-Gaussian factors. The root mean square error (RMSE) is the widely accepted standard metric for evaluating filter performance in the literature. It measures the standard deviation of the differences between estimated and true states, indicating the filter's accuracy. A lower RMSE value signifies better performance, meaning that the filter is estimating true states more accurately.

1.5 Thesis objective

The author outlines the following objectives for this thesis, based on the motivations of the thesis discussed above.

- To develop an advanced Gaussian filtering algorithm for handling delayed measurements with a reduced number of prerequisite delay probabilities.
- To design a Gaussian filtering algorithm for handling two measurement irregularities together, particularly the delayed measurements and the non-Gaussian measurement noises.
- To introduce a nonlinear Gaussian filtering algorithm for handling cyber-attacked measurement data.
- To extend the traditional particle filtering methodology for angular data.
- To validate the improved accuracies of the proposed method for a couple of popular nonlinear filtering problems taken from the literature.

1.6 Approaches and Methods

As already indicated, the objective of this thesis is to develop several nonlinear filtering methods, as extensions of the traditional Gaussian and particle filtering methodologies, to handle different forms of data irregularities. In developing any new algorithm, the author takes the following approaches.

- Model formulation: The concerned data irregularity problem is formulated mathematically for each development. In this regard, the author generally introduces the mathematical model that incorporates the effects of the data irregularities. However, a somewhat different approach is taken in handling the angular data. In this case, the author formulates the problem by amending the general structure of the unknown PDF, requiring a different numerical approximation method.

- Filter design: In this step, the traditional filtering methods are re-derived for the mathematically formulated problem statement to develop advanced Gaussian filtering algorithms. In the case of angular data, a new numerical approximation method is introduced.

- Validation: The developed filtering algorithms are simulated for a couple of nonlinear filtering problems. The simulations are performed in MATLAB over a personal computer with the 64-bit operating system, 32 GB RAM, and configuration Intel i5, 3.40GHz processor.

- Performance criteria: The author uses the performance metrics as the root mean square error (RMSE), average RMSE (ARMSE), and computational time to validate the performance of developed filters in this thesis.

1.7 Contributions

The main contributions of this thesis are summarized below.

- This thesis provides a detailed and up-to-date literature review on nonlinear filtering, including the contributions available for handling various measurement irregularities.

- A nonlinear Gaussian filtering algorithm is developed for handling delayed measurements with a reduced number of prerequisite information about the probabilities of delay.

- The traditional Gaussian filtering is re-derived for handling the delayed measurements under non-Gaussian noisy measurements.
- An advanced Gaussian filtering algorithm is introduced for handling cyber-attacked measurements.
- The traditional particle filtering method is redesigned for efficiently handling with angular data.
- The simulation results provide and validate its improved accuracies for the aforementioned nonlinear filtering algorithms.

1.8 Publications generated out of thesis work

Journal papers:

1. Guddu Kumar, Sumanta Kumar Nanda, Alok Kumar Verma, Vimal Bhatia, and Abhinoy Kumar Singh, "Nonlinear Gaussian Filtering with Network-Induced Delay in Measurements," in IEEE Transactions on Aerospace and Electronic Systems, 2022. doi:10.1109/TAES.2022.3182936.
2. Guddu Kumar, Venu Gopal Yamalakonda, Swaminathan R, and Abhinoy Kumar Singh, "Fractionally Delayed Bayesian Approximation Filtering under Non-Gaussian Noisy Environment" in IEEE Transactions on Aerospace and Electronic Systems, 2023. doi: 10.1109/TAES.2023.3266176.
3. Guddu Kumar, Sumanta Kumar Nanda, Swaminathan R, and Abhinoy Kumar Singh, "Gaussian Filtering with Cyber-Attacked Data" in IEEE Transactions on Aerospace and Electronic Systems, 2022. (Under Revision).
4. Guddu Kumar, Amit Kumar Naik, Ram Bilas Pachori, Swaminathan R, and Abhinoy Kumar Singh, "Improved Gaussian Filtering for Handling Concurrent Delayed and Missing Measurements" in Asian Journal of Control, 2022. (In press).
5. Guddu Kumar, Paresh Date, Ram Bilas Pachori, Swaminathan R, and Abhinoy Kumar Singh, " Wrapped Particle Filter for Angular Data",in IEEE Access, vol. 10, pp. 90287-90298, 2022, doi: 10.1109/ACCESS.2022.3200478.

Book Chapters

1. Guddu Kumar, Vikas Kumar Mishra, Swaminathan R, and Abhinoy Kumar Singh, "Parameter Identification of Coulomb Oscillator from Noisy Sensor Data", in Communication and Control for Robotic Systems 2022 (pp. 327-338). Springer, Singapore.

Conferences:

1. Guddu Kumar, Swaminathan R, and Abhinoy Kumar Singh, "High-degree Cubature Quadrature Kalman Filter with Fractional Delayed Measurement", IEEE 19th India Council International Conference (INDICON) 2022, Nov 24 (pp. 1-6).

1.9 Thesis Organization

The rest of the thesis is structured as follows. The thesis consists of seven chapters including the present one. Continuing from this chapter, the second chapter provides a brief overview of the remarkable developments in the field of nonlinear estimation and filtering. It is followed by a brief discussion of the Bayesian framework of nonlinear filtering. Chapters 3 and 4 contribute to the estimation with delayed measurements. In Chapter 5, a modified framework of filtering is proposed for dealing with the arbitrary nature of cyber-attacked measurements. In Chapter 6, an algorithm is developed for estimating the states of a discrete system with bounded data. The last chapter briefs the discussions and conclusions of the thesis. Moreover, it includes the scope of future works. In the end, an appendix is provided which includes the algorithms developed in different chapters.

Chapter 2

Literature Review

As discussed in the previous chapter, this thesis aims to develop advanced filtering algorithms for discrete-time nonlinear systems. Therefore, the review, in this chapter, is limited to the filtering algorithms designed for discrete-time nonlinear state-space models.

Among various discrete-time nonlinear filtering algorithms available in the literature, this review broadly focuses on the computationally efficient Gaussian filtering, which is a Bayesian approximation filtering method. The review also briefly covers the particle filtering, which is another popular nonlinear filtering methodology. Furthermore, the review includes the extensions of the traditional Gaussian filtering algorithms for addressing various irregularities in measurements, such as delayed measurements, missing measurements, and cyber-attacks.

2.1 Gaussian Filtering

The previous chapter introduced two Gaussian filtering approaches, namely the derivative-free Gaussian filtering and derivative-based Gaussian filtering.

As discussed previously, the derivative-based Gaussian filtering is an extension of the linear Kalman filter with a locally linearized nonlinear function. The popularly known extended Kalman filter (EKF) [26] and its extensions are the prevalent developments under this approach.

In the derivative-free Gaussian filtering, the intractable integral is numerically approximated using a set of sample points and associated weights. The literature witnesses various derivative-free Gaussian filters using various numerical approximation methods, giving different sets of sample points and weights. The derivative-free Gaussian filters outperform the derivative-based Gaus-

sian filters in terms of accuracy and numerical stability.

To this extent, we review the various developments under the derivative-free Gaussian filtering and the derivative-based Gaussian filtering. This chapter also discusses another extension of Gaussian filtering named Gaussian-sum filtering.

2.1.1 Derivative-based Gaussian filters

As discussed above, the popular contributions under the derivative-based Gaussian filtering include the EKF and its variants, which are reviewed below.

EKF and its variant

The EKF was developed in the sixties [26], within a few years after the development of the linear Kalman filter. The EKF is probably the earliest developed nonlinear Gaussian filter, which is still widely popular in practical applications as well as in academic developments. As discussed previously, the EKF locally linearizes the nonlinear dynamic models by computing the first-order derivative in terms of the Jacobian. Subsequently, it applies the concept of linear Kalman filtering over the locally linearized nonlinear dynamical models.

The EKF traditionally suffers from several drawbacks. For example, for the derivative computation, it requires the system dynamics to be continuous. Similarly, the first-order linearization of the nonlinear dynamical systems causes poor accuracy and numerical instability. This problem becomes severe particularly if the sampling interval is large. Please note that the sampling interval is often system/device property, which cannot be flexibly commanded by the practitioners to mitigate these drawbacks. Nevertheless, irrespective of these drawbacks, the EKF widely attracts the practitioners due to its simplicity of implementation and small computational demand. Moreover, many variants of the traditional EKF are developed to mitigate its drawbacks up to some extent. In the following discussion, the author reviews some of the popular variants of the EKF.

- Second-order extended Kalman filter [27]: The EKF traditionally linearizes the nonlinear dynamics by computing the Jacobian, giving a first-order approximation of the nonlinear dynamics in its Taylor-series expansion. The SEKF reformulates the traditional EKF filtering structure to accomplish the second-order approximation of nonlinear dynamics. Subsequently, it improves the filtering accuracy.

- High-order extended Kalman filter [28]: The high-order EKF is a further generalization of the EKF and second-order EKF, accomplishing higher-order approximation of the nonlinear dynamics. Subsequently, it further improves the filtering accuracy.

- Modified gain extended Kalman filter [29]: In a modification to the conventional EKF, it is introduced by Song *et al*. where the gain appears in terms of past measurements.

Irrespective of the various variants of the EKF available in the literature, the derivative computation remains intact and a major worry. As discussed previously, the derivative-free Gaussian filtering method could mitigate the drawbacks of the EKF and its variants up to some extent. In the subsequent discussion, the author reviews the popular developments under the derivative-free Gaussian filtering.

2.1.2 Gaussian-Newton filter

The Gauss-Newton (GN) [30] filter is a class of derivative-based filters used to estimate the state of a nonlinear system. It is based on the minimum variance theory and is unbiased. The GN filter can be applied in both recursive and non-recursive forms [31].

One of the advantages of the GN filter is that the filter can perform measurement updates at any time stamp, allowing for long-term prediction. Additionally, the filter has a finite memory length that can be adapted to estimate highly changing dynamics. The memory length acts like a sliding window and allows the filter to focus on the most recent observation, minimizing the effect of previous dynamics on the current estimation [32].

Initially, the GN filter was derived by assuming deterministic dynamics, but it has been shown that it can easily be adapted for processes with dynamic noise. The filter's finite memory nature provides the best means of mitigating the effect of the process noise covariance matrix, which requires fine-tuning in the Kalman filter. The adapted filter explains the error covariance inconsistency of the Kalman filter. Another advantage of the GN filter is that it does not require initialization since the covariance matrix is computed in accordance with the available batch of data [33].

In state estimation, the GN filter estimates the state of a system by iteratively minimizing the difference between predicted and actual measurements. The predicted measurements are calculated using the measurement function and the current estimate of the state vector, which relates

the state vector to the measurements and is often nonlinear. The GN filter assumes that the measurement noise is Gaussian and that the measurement function is differentiable. The algorithm continues iterating until a termination criterion is met, such as a maximum number of iterations or a small change in the state estimate between iterations. Overall, the Gauss-Newton filter is a powerful algorithm for estimating the state of nonlinear systems with Gaussian measurement noise. However, it may not be suitable for systems with non-Gaussian noise or non-differentiable measurement functions.

2.1.3 Derivative-free Gaussian filters

As discussed in the previous chapter, the derivative-free Gaussian filtering involves intractable integrals, which are numerically approximated during the filtering. The literature witnesses various developments using different numerical approximation methods. In follow-up discussions, the author reviews the major developments in the derivative-free Gaussian filtering.

Unscented Kalman filter (UKF) and its variant

The UKF [34], developed in the nineties, is probably the first derivative-free alternative of the derivative-based EKF and its variants. As a general derivative-free Gaussian filter, the UKF encounters intractable integrals during the filtering. It utilizes an unscented transformation-based numerical approximation method [35], which approximates the intractable integrals using a set of sigma points and weights. Being a derivative-free solution, the UKF improves the accuracy and numerical stability in comparison to the EKF.

Over the time, several variants of the UKF appeared in the literature. This thesis provides a summary of a few developments in the following discussions.

- Scaled unscented Kalman filter [36]: The scaled unscented Kalman filter generalizes the traditional method of generating the sigma points and weights by introducing some scaling parameters. This could help in improving the accuracy further without increasing the computational demand.

- Square-root unscented Kalman filter [37]: The UKF conventionally requires computing the Cholesky decomposition of covariance matrices, giving the square-root of the covariance

matrices. However, computing the Cholesky decomposition requires the covariance matrices to be positive-semidefinite. On the other hand, in practical applications, the positive-semidefinite is often lost, resulting in the failure of the filter implementation. To preclude this limitation of the UKF, and to ensure consistency in filter execution, the square-root unscented Kalman filter is introduced. The square-root unscented Kalman filter precludes the computation of the Cholesky decomposition by utilizing QR decomposition.

- Adaptive unscented Kalman filter [38]: The UKF is conventionally implemented by assigning the noise parameters dubiously. However, an improper noise selection often results in poor accuracy or a complete failure of the filter. This drawback is mitigated in the adaptive unscented Kalman filter, which is adaptive to noise variations.

- Transformed unscented Kalman filter [39]: In the transformed unscented Kalman filter, the sigma points are orthogonally transformed without changing the weights. The orthogonally transformed sigma points reduce the higher-order moments in computing the estimate and covariance. Thus, the orthogonally transformed sigma points reduce the estimation errors. Subsequently, the transformed unscented Kalman filter outperforms the traditional UKF without increasing the computational demand.

Some other popular extensions of the UKF are simplex sigma set UKF [40], spherical simplex sigma set UKF [41], marginalised iterated UKF [42], risk-sensitive UKF [43], and new sigma point Kalman filter [44]. Moreover, [45],[46],[47] highlight some practical applications of the UKF. More specifically, [45] applies the UKF for the state of charge estimation in an adaptive cell model, [46] implements the UKF for estimation of dynamic state for multi-machine power system, and [47] applies the UKF for the biomedical system.

Cubature Kalman filter (CKF) and its variant

In 2009 by Arasaratnam *et al.* [48] introduced another popular Gaussian filter named cubature Kalman filter by utilizing a different numerical approximation method. The CKF decomposes the intractable integral into spherical and radial integrals. Subsequently, the spherical integral is approximated using the third-degree spherical-cubature rule, while the radial integral is approximated using the first-order Gauss-Laguerre quadrature rule. The combination of the third-degree spherical-cubature rule and the first-order Gauss-Laguerre quadrature rule is called

the third-degree spherical-radial rule. Finally, the author can state that the CKF approximates the intractable integral using the third-degree spherical-radial rule.

The CKF utilized lower orders of the spherical-cubature rule and Gauss- Laguerre quadrature rule, giving poor approximation accuracy for the spherical and radial integrals. Thus, it leaves the scope for further improvement by advancing the orders of the spherical-cubature rule and Gauss- Laguerre quadrature rule. As a result, a series of developments appeared by improving the orders of two numerical approximation rules. This thesis briefly reviews some major developments below.

- Cubature quadrature Kalman filter (CQKF) [49]: The CQKF is a generalization of the CKF to improve the estimation accuracy with a marginally increased computational demand. It adopts the integral decomposition strategy directly from the CKF. It further retains the third-degree spherical cubature rule from the CKF for approximating the spherical integral. However, it utilizes the higher-order Gauss-Laguerre quadrature rule for approximating the radial integral. By combining the third-degree spherical cubature rule and the higher-order Gauss-Laguerre quadrature rule, the CQKF formulates a new numerical approximation method named the cubature quadrature rule. The sample points generated by this rule are known as cubature quadrature points. The use of the higher-order Gauss-Laguerre quadrature rule for approximating the radial integral causes the improved accuracy of the CQKF. Moreover, the computational demand of the higher-order Gauss-Laguerre quadrature rule is relatively higher than the first-order Gauss-Laguerre quadrature rule, resulting in a higher computational demand of the CQKF in comparison to the CKF.

- Higher-degree cubature Kalman filter (HDCKF) [50]: Similar to the CQKF, the HDCKF is another popular generalization of the CKF. This generalization is very much similar to the CQKF, except that it modifies the numerical approximation method of spherical integral instead of the radial integral. The HDCKF replaces the third-degree spherical cubature rule (used in the CKF) with the higher-degree spherical cubature rule. As a result, it improves the accuracy in comparison to the CKF. Although, the use of the higher-degree approximation of the spherical integral comes at the cost of increased computational demand in comparison to the CKF.

- Higher degree cubature quadrature Kalman filter (HDCQKF) [51]: The HDCQKF is a

further generalization, which utilizes higher-order of approximation for both the radial and spherical integrals. More specifically, it utilizes the higher-degree spherical cubature rule for approximating the spherical integral, while it utilizes the higher-order Gauss-Laguerre quadrature rule for approximating the radial integral. Thus, its accuracy is better than each of the CKF, CQKF, and HDCKF. However, its computational demand is also higher than the CKF, CQKF, and HDCKF.

Some other popular extensions of the CKF are the square-root cubature Kalman filter [48], square-root cubature quadrature Kalman filter [52], transformed cubature quadrature Kalman filter [53], simplex-spherical cubature Kalman filter [54], simplex-spherical cubature quadrature Kalman filter [55], and exponential-fitted cubature Kalman filter [56]. Furthermore, the widespread practical application of the CKF and its extensions are reflected from [57],[58],[59]. For example, [58] used the CKF in underwater target tracking applications, [57] implemented the CKF for continuous glucose monitoring, while [59] performed fault diagnosis using the CKF.

Gauss-Hermite filter (GHF) and its variant

The GHF [60] utilizes the univariate Gauss-Hermite quadrature rule for numerically approximating the intractable integrals. In this filter, the sample points generated for approximating the intractable integrals are often known as quadrature points. As the practical problems are mostly multivariate, the GHF utilizes product rule [60] for extending the univariate Gauss-Hermite quadrature rule into a multivariate domain. However, for the product rule, the number of quadrature points increases exponentially with the increasing system dimension [61]. Therefore, the GHF suffers from the curse of dimensionality problem and becomes inapplicable for high-dimensional systems. To reduce the computational demand, there are two popular variants of the GHF available in the literature, which are discussed below.

- Sparse-grid Gauss-Hermite filter (SGHF) [61]: The SGHF replaces the product rule with the Smolyak rule, reducing the number of multivariate quadrature points. Interestingly, it reduced the computational demand significantly without damaging the accuracy.

- Adaptive sparse-grid Gauss-Hermite filter (ASGHF) [62]: The ASGHF utilizes the adaptive-sparse grid method of extending the univariate quadrature rule into a multivariate domain.

The adaptive-sparse grid method considers varying nonlinearity across different dimensions. Conversely, it reduces the number of multivariate quadrature points further. As a result, the ASGHF reduces the computational demand further in comparison to the GHF and SGHF, without harming the accuracy.

Some other popular contributions in the GHF-based filtering are square-root Gauss-Hermite filter [63], generalized GHF [64], and multi-sparse grid GHF [65]. Similar to the square-root unscented Kalman filter, the square-root Gauss-Hermite filter precludes the Cholesky computation. The generalized GHF improves the accuracy further. Meanwhile, the multi-sparse grid GHF reduces the computational demand further, considering that some of the subspaces of the unknown states are uncorrelated.

The GHF and its variants are among the most accurate Gaussian filters available in the literature. Despite of the reduced computational demands for filtering applications in general the SGHF and the ASGHF, their computational demand is still too large for high-dimensional systems, precluding their practical applications widely.

2.1.4 Gaussian-sum filter

As discussed previously, the derivative-based as well as the derivative-free Gaussian filters approximate the unknown PDFs as Gaussian. In many cases, the Gaussian approximation may be a significantly poor characterization of the unknown PDFs, resulting in poor accuracy. Although this concern is remarkably taken care of in the particle filtering, its computational demand is very large. The Gaussian-sum filtering provides a trade-off between the Gaussian filtering and particle filtering in terms of computational demand as well as accuracy. Unlike the traditional Gaussian filtering, the Gaussian-sum filtering [66] characterizes an unknown PDF with the help of multiple Gaussian PDFs instead of a single Gaussian. Its accuracy increases as the number of Gaussian PDFs increases. At the same time, the computational demand also increases with the increasing number of Gaussian PDFs. The literature witnesses various contributions, such as the Gaussian-sum extended Kalman filter [67], Gaussian-sum unscented Kalman filter [68], Gaussian-sum cubature Kalman filter [69], and Gaussian-sum Gauss Hermite filter [70] to redesign the traditional Gaussian filters under the Gaussian-sum extensions.

2.2 PF and its variant

The particle filtering was originally introduced in 1993 by Gordon *et al.* [20]. However, its background starts from eighties itself, when the Monte-Carlo method (MCM)-based approximation of the desired conditional PDFs became popular [71]. Interestingly, the MCM-based filtering does not make particular assumptions and approximations for the unknown PDFs and the noises, unlike traditional Gaussian filtering. Subsequently, the particle filtering is free from such assumptions and approximations, resulting into, in general, very high accuracy.

As discussed previously, due to the large computational demand, the particle filtering is often ignored in practical applications. In order to reduce the computational demand, the literature witnesses various resampling techniques that can be used for ignoring the negligibly-weighted particles and reducing the number of particles [72],[73],[74]. Arulampalam *et al*. [75] reviewed various particle filtering methods designed with different resampling techniques. Furthermore, the development of sophisticated, efficient, and low-cost computational devices, in recent years, has attracted the practitioners for applying the particle filtering in practice applications. For example, [76] applied particle filtering in target-tracking problems, while [77] applied particle filtering for monitoring the health of industrial equipment.

The selection of an appropriate proposal density is crucial for particle filters. To ensure this, the literature integrated the general Gaussian filters in the particle filtering framework for computing the proposal density. For example, the unscented particle filter (UPF), cubature particle filter (CPF), and Gauss-Hermite particle filter (GHPF) are designed in [78], [79], and [80], respectively, by integrating the UKF, CKF, and GHF, respectively, in the particle filtering framework. Some other popular extensions of the particle filtering are Rao-Blackwellized particle filtering [81], likelihood particle filter [82], auxiliary particle filter [83], and Gaussian-sum particle filter (GPF) [84].

2.3 Filtering with various data irregularities

The practical measurement data suffer from various irregularities, which are ignored in the above-reviewed filtering methods. Apparently, the above-reviewed filters often suffer from poor accuracy, when the measurement data consist of one or more irregularities. Some of the popu-

lar measurement irregularities are delayed measurements, missing measurements, non-Gaussian measurement noises, and cyber-attacked measurements. In another problem, the above-discussed filters are designed for infinite-range data, while the practically common angular data are of finite-range, particularly, in the range of $[-\pi, \pi)$. The author considers the angular data also as a problem of data irregularity. In the subsequent discussions, the author briefly reviews the various developments introduced for handling each of these irregularities.

2.3.1 Delayed measurements

There are various sources of delay in measurements, including the propagation delay and queuing delay [85],[86],[87]. The delayed measurement carries misleading information about the current state, resulting into poor accuracy of the traditional filters. In many cases, the delay is known and the problem can be trivially resolved with a simple time-shift. Meanwhile, in many cases, particularly in the lack of time-stamping and access to clocks, the delay becomes unknown. In the literature, the unknown and time-varying delay is commonly known as random delay. This thesis is particularly concerned with random delays. The randomly delayed measurements commonly occur in dynamics control [88],[89], aerospace target tracking [90], communication systems [91], *etc*.

The nonlinear filtering literature on delayed measurements mainly begins from the contribution of Hermoso-Carazo *et al.* [92] on handling restricted delays. In this publication, the traditional EKF and UKF were extended for handling one-delay, *i.e.*, the delay being up to one sampling interval. This filtering method was later separately extended for the CKF [93] and for the GHF [94]. The same filtering approach was extended in [95] for handling two-delays, *i.e.* the instantaneous delay being up to two sampling intervals. In another similar contribution, Zhao *et al.* [96] extended the one-delay filtering algorithm for correlated noises. Although these algorithms could open a new domain of research, they were restricted to only small delays, while the practical problems often encounter larger delays. With this motivation, Singh *et al.* [97] extended the traditional Gaussian filtering algorithm for handling any large delays.

The above-mentioned nonlinear Gaussian filtering literature for handling delayed measurements requires the delay models available. The formulation of delay models requires precise information about the probabilistic information of the delay. However, very often, the delay is

completely uncertain and no such information is available. For filtering under this scenario, [98] introduced a likelihood-based delayed filtering algorithm. In this algorithm, a likelihood-based delay identification is used, which helps in restructuring the filtering algorithm according to the identified delay. In another contribution, Esmzad *et al*. [99] used the likelihood-based approach for determining the probabilistic information about the delay. Following this, it implemented the delay model-based filtering approach.

2.3.2 Missing measurements

In practice, the measurements are often intermittently missing [100],[101]. The intermittently missing measurements may occur due to various reasons. For example, the network and communication channels [102] used in measurement data transmission may have packet losses, causing intermittently missing measurements. Similarly, very often, the sensor temporal failure [103] and the use of time-sharing sensors [104] cause the intermittently missing measurements. Furthermore, the measurement data may be often unidentifiable due to high noise and clutter [105]. In some cases, this scenario is also formulated as a problem of missing measurements.

In [106], authors had modified the traditional EKF to filter with partial missing measurements and an uncertain system, when only a fraction of the measurements are available. Following this, [107] and [108] are utilized the UKF for nonlinear stochastic systems with missing measurements. Further, [109] developed a distributed filtering method for saturated systems. Nevertheless, it focuses on linear dynamical models with a few modifications for saturation-induced nonlinearities. A recent contribution [110], addressed the joint occurrence of delayed and missing measurements by tuning of large number of delay probabilities.

2.3.3 Non-Gaussian measurement noises

The previously reviewed nonlinear Gaussian filters are designed with Gaussian approximation of arbitrary noises. However, the practical measurement noises are inherently non-Gaussian and their Gaussian approximation often significantly harms the estimation accuracy [2].

The traditional Gaussian filters underperform for non-Gaussian noises mainly because they are designed under the minimum mean square error (MMSE) criterion, which ignores the higher-order moments (beyond the second-order) of the non-Gaussian noises [19]. In the recent year litera-

ture, the information-theoretic learning (ITL)-based methods are gaining popularity in handling non-Gaussian noises [111], [112]. Such methods define particular cost functions and optimize them to extract higher-order statistics. As they extract high-order statistics, they are accurate for non-Gaussian noises. In general, the two popular cost functions are derived in terms of correntropy [113] and entropy [114, 115], while some other statistical measures also exist with lesser popularity [116, 117]. The literature, however, has favored the use of the correntropy-based cost functions because of lower computational complexity compared to the entropy-based cost functions [118]. Consequently, the author also adopts a correntropy-based cost function. Correntropy is a measure of similarity between two random variables within a certain neighborhood of a joint space, controlled by a fixed window. The fixed window is often known as kernel bandwidth. A higher-order similarity is indicative of larger correntropy. Alternatively, correntropy maximization ensures the best similarity between two random variables. Thus, we design our filter under the maximum correntropy (MC) criterion in order to handle the non-Gaussian noises. The literature on MC criterion-based filtering starts from [119], which redesigns the linear Kalman filter under this criterion. Later, various contributions [120],[121],[122],[123],[124],[125] appeared by redesigning the popular nonlinear Gaussian filters, such as the EKF, UKF, CKF, and GHF, under this criterion.

In the recent years, some other design criteria have also been tested for handling non-Gaussian measurements. For example, Huber-based cost function [126] and minimum entropy criterion-based [127],[128] design have already been tested in the filtering literature. Furthermore, these design criterion utilizes kernels. Although the use of Gaussian kernel [129],[130] is probably the most popular, the filtering literature exhibits contributions with other kernels as well [131].

2.3.4 Cyber-attacked measurement

The filtering accuracy is largely influenced by the measurement's precision, which is often deliberately tampered and malformed through cyber-attacks [132],[133]. In the cyber-attacks, an intruder may tamper the true measurements with the following irregularities: i) false data injection (FDI): injecting false data alongside the true measurements [134], ii) time asynchronous measurements (TAM): introducing time-delay in measurement propagation [92], and iii) denial-of-services (DoS): denies any measurement availability [135],[136].

Please note that, in general, an intruder attacks the data only intermittently to keep the attack unrecognizable. Considering that the attack occurs intermittently, the TAM attack and DoS attack cause intermittently occurring delayed and missing measurements, respectively. Interestingly, as discussed above, the literature witnesses various developments for handling the intermittently occurring delayed and missing measurements individually [92],[95],[100]. Although these developments are not designed in the context of handling cyber-attacks, they can be efficiently used for handling TAM and DoS attacks.

For nonlinear filtering under the FDI attack, the literature witnesses only a few preliminary developments. For example, [134] redesigns the EKF for addressing the FDI attacks, while [137] re-derives the UKF for filtering under the FDI attack. These developments are restricted to additive false data, mimicking false data independent of the true measurements. A very recent work [138] develops two algorithms, one for handling the additive false data, while another for handling the multiplicative false data. Alternatively, it introduces two algorithms; the first algorithm handles the false data that is independent of the true measurements, while the second algorithm handles the false data if it is dependent on the true measurements. Following a similar idea, another development [139] has appeared recently to handle both the additive and multiplicative false data in a single algorithm.

2.3.5 Angular Data

There is a dearth of literature for handling nonlinear filtering problems with angular data. Nevertheless, some developments appeared to tackle this problem using filtering approaches beyond the Gaussian filtering and particle filtering methodologies. In a preliminary development, [140] utilized truncated Fourier series with wrapped normal (WN) distribution for addressing the angular data. However, the finite-length truncation of the Fourier series severely harms the estimation accuracy. Some of the later developments [141],[142],[143] are particularly designed for linear system models and fail for nonlinear systems. In another development, Kurz *et al*. introduced a nonlinear circular filtering method through a couple of publications [144],[145],[146],[147]. Specifically, this method is particularly designed for univariate systems, though most of the real-life filtering problems are multivariate.

2.4 Summary

We summarize the detailed literature review provided in this chapter through the following notes.

- The existing nonlinear filters are sub-optimal. Thereby, a variety of filtering methods are available for achieving different accuracy levels at the cost of increasing the computational budget.

- The nonlinear filters traditionally ignore various data irregularities, in general, the measurement data irregularities. Such irregularities often decisively appear in practical applications and harm the accuracy. Some of the popular measurement irregularities are delayed and cyber-attacked measurements.

- The nonlinear filters are designed for standard state space models of nonlinear dynamical systems, which are formulated with various simplifications. For example, the state and measurement variables are considered to be of infinite-range and the noises are assumed to be Gaussian. The practical problems often significantly deviate from such simplifications, which causes poor accuracy of the existing filters. In the remaining chapters, such problems are also treated as the problem of data irregularity.

- The literature witnesses various developments for handling different measurement irregularities individually. Nonetheless, in practice, two or more measurement irregularities may appear simultaneously, which can harm the accuracy.

Chapter 3

Gaussian Filtering with Delayed Measurements

3.1 Introduction

As discussed in the previous chapters, the Gaussian filters are conventionally designed for non-delayed measurements, *i.e.* a measurement received by the estimator at time t_k must be generated at the same time t_k. Yet, in practice, the measurements are commonly delayed due to various reasons.

To illustrate the presence of delay in the measurement data, let us consider the example of the network-induced delay [148],[149]. In the modern estimation and filtering applications, the measurements are commonly transmitted through network systems and multiplexed communication channels [86], [150]. The network systems and multiplexed communication channels often induce time-delay in measurements [92], [95], [93], [151]. Due to the time-delay, the measurement generated at $t_k - t_d$ reaches the estimator at t_k with t_d delay. Subsequently, the delayed measurements convey misleading information about the states and deteriorate the estimation accuracy of the traditional Gaussian filters that are designed for non-delayed measurements.

This chapter is specifically focused on random delays, where the delay is unknown and time-dependent. Such delays occur if time stamping and synchronization are difficult in the measurement data transmission. The filtering problems with randomly delayed measurements frequently appear in network systems [148], [152], target tracking [153], communication system [5], *etc*.

We reviewed the various contributions for handling the delayed measurements in Section 2.1.3 of Chapter 2. In the detailed review, we discussed that the early developments [92], [95], [93], [151] could handle only small delays. Meanwhile, in the recent contributions [6],[97],[98],[99] are robust for large delays as well. Interestingly, the recent contributions suffer from the following drawbacks: i) Refs. [97],[6] require a large number of delay probabilities and their accuracy broadly depends on the precision of selecting the delay probabilities. Conversely, such delay probabilities are mostly unknown or difficult to identify. Therefore, as the number of the delay probabilities increases, the chances of improper selection or identification of delay probabilities increase. Consequently, as the number of delay probabilities increases, the accuracy of the filter degrades. ii) Refs. [98], [99] usually select an upper bound of the delay arbitrary. An underestimated upper bound causes intermittent measurement losses, which harms the accuracy [87]. On the other hand, an overestimated upper bound allows the randomly chosen delay to be considerably higher than the actual delay, which further harms the accuracy. In summary, the arbitrary selection of the upper bound of delay essentially contributes to estimation error.

This chapter introduces a modified Gaussian filtering method for handling large delays in measurements. The proposed filtering method reduces the required number of delay probabilities, resulting into, in general, an improved accuracy. Additionally, it precludes the ambiguity of upper bound selection, which further leads to an improved accuracy. The proposed filtering algorithm is also extended for multiple sensor scenarios to allow different delays for different elements of measurement. The proposed algorithm is based on a modified measurement model, which is the stochastic summation of the current and past measurements. The stochastic modeling is based on Bernoulli and geometric random variables. The Bernoulli random variable represents whether the measurement is delayed or not. If a delay occurs, the geometric random variable represents the extent of delay to decide which of the past measurements is received. It should be mentioned that the geometric distribution is probably used for the first time in delay filtering literature, which could preclude the need of various ambiguous delay probabilities. Based on the modified measurement model, the traditional filtering method is redesigned to address the problem of delayed measurements. Thus, the author derives the expressions of measurement estimate, covariance, and cross-covariance between the state and measurement. The derivation is based on the modified measurement model.

In conclusion to the above discussion, the major contributions and novelties of this chapter are

as follows:

- To the best of the author's knowledge from the literature review, the proposed method for the first time models the delayed measurements using the geometric distribution, which has several advantages over the previously used Bernoulli distribution.
- The delay modeling strategy used in this chapter requires only two probabilities to be known, unlike the existing delay filters that require many more delay probabilities in handling large delays.
- This chapter redesigns the classical Gaussian filtering for handling delayed measurements by re-deriving the expressions of the estimate and covariance of measurement, and the cross-covariance between the state and measurements.
- This chapter simulates two nonlinear filtering problems with delayed measurements and provides a comparative performance analysis of the proposed and existing methods.

Note that the proposed filtering technique can be implemented with any of the existing Gaussian filters, like the EKF[26], UKF[34], CKF [48], and GHF [60]. The author chooses the CKF to test the performance due to its popularity in offering high accuracy at low computational demand. The proposed filtering approach outperforms the traditional CKF in the presence of delayed measurements. It also outperforms the extended CKF with the existing delayed filtering techniques.

3.2 Problem Formulation

Recalling Chapter 1, the state space model for representing a general nonlinear dynamical system can be given as

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

$$\mathbf{z}_k = \Psi_k(\mathbf{x}_k) + \eta_k. \tag{3.2}$$

Please refer to Chapter 1 for a detailed discussion, including the clarifications on the notations.

Please note that the above state space model represents a non-delayed system. Therefore, $\mathbf{z}_k$ must be a non-delayed measurement and may carry misleading information about the states in the presence of delay. As discussed in the previous section, the popular delay filters [97],

[99] suffer from two major drawbacks: i) they require prior knowledge of a large number of delay probabilities and ii) they arbitrarily choose an upper bound of delay. These drawbacks appear mainly due to the improper delay modeling strategy used by them. Thus, to handle these drawbacks, it is required to reconstruct the delay modeling strategy and redesign the traditional filtering methodology accordingly.

It should be mentioned that the geometric distribution can characterize the time-varying multiple delay possibility with just one probability information, unlike the traditionally used sequence of Bernoulli random variables. Moreover, the geometric random variable also mitigates the need to fix the upper bound of delay arbitrarily. As the measurements are more often non-delayed, the author may like to have a different approach for the non-delay situation. Thus, the author continues using a Bernoulli random variable for modeling a non-delay scenario.

Besides the geometric distribution, other distributions also appear with reduced numbers of delay probabilities, *e.g.*, Binomial distribution, and gamma distribution [154]. More importantly, the uniqueness of delay is suitably characterized by sequential Bernoulli trials [6],[97], where the past instants are examined for coinciding with the current measurement's arrival time. In this context, the geometric distribution is possibly more suitable.

Following the above discussion, the author reformulates the measurement model using a Bernoulli random variable β_k and a geometric random variable $G_{d,k}$ as

$$\mathbf{y}_k = \beta_k \mathbf{z}_k + (1-\beta_k) \sum_{i=1}^{k-1} G_{d,k}(i) \mathbf{z}_{k-i}, \tag{3.3}$$

where $\mathbf{y}_k$ represents the measurement with delay possibilities and d stands for representing the delay extent. This expression also represents a unique selection strategy for $\mathbf{y}_k$ to appropriately pick an element out of several time-synchronized hypothetical past measurements $\mathbf{z}_{k-i}$ using β_k and $G_{d,k}$. Please note that $\beta_k = 1$ denotes a non-delayed measurement at t_k, while $\beta_k = 0$ represents a delay occurrence.

Assumption 3.2.1 *The random variables are assumed to be independent to each other.*

Let us illustrate the delay model presented in (3.3). If the delay occurs (*i.e.*, $\beta_k = 0$), $G_{d,k}(i)$ represents the i^{th} delayed instant from which a measurement arrives. Let the geometric random variable is denoted as $G_{d,k}$, which indicates the measurement received at the t_k instant that is unknown with d-step delay, where $d \in \{1, 2, \cdots, k-1\}$. At any instant t_k, $G_{d,k}$ is of size d,

with the first $d-1$ elements being zero and a one at the d^{th} element to denote the arrival of measurement from t_{k-d}. The author considers a varying delay scenario, *i.e.*, the size of $G_{d,k}$ varies for every k, *e.g.*, $G_{d,k} = (1)$ for one delay and $G_{d,k} = (0,1)$ for two delay. However, to maintain a consistent length of the array $G_{d,k}$ as $k-1$, the author assigns zeros into the remaining elements, *i.e.*, $G_{d,k} = (\mathbf{0}_{d-1}, 1, \mathbf{0}_{k-d-1})$, where $\mathbf{0}_d$ represents an all-zero array of size d. Note that this $G_{d,k}$ does not imply that 1 can only occur at t_k, as it can occur at specific d. The author simply restricted the maximum delay at t_k to be $k-1$.

Fig. 3.1 represents the arrival of a delayed or a non-delayed measurement for various values of β_k and $G_{d,k}$. $\beta_k \in \{0,1\}$ satisfies the following properties,

$$\begin{aligned} &P(\beta_k = 1) = \mathbb{E}[\beta_k] = \mathbb{E}[(\beta_k)^m] = p_b \\ &P(\beta_k = 0) = \mathbb{E}[1-\beta_k] = \mathbb{E}[(1-\beta_k)^m] = q_b = 1-p_b \\ &\mathbb{E}\left[(\beta_k - p_b)^2\right] = \mathbb{E}\left[(\beta_k)^2\right] - \mathbb{E}[\beta_k]^2 = p_b(1-p_b), \end{aligned} \tag{3.4}$$

where $m \in \mathbb{R}$ is a constant. The author uses the notation G^j to denote a $(k-1)$-dimensional $G_{d,k}$ with j^{th} element being 1. Further, it denotes the probability of 1 at each element of $G_{d,k}$ as p_g. Subsequently, if $G_{d,k}(i)$ denotes the i^{th} element of $G_{d,k}$, then

$$\begin{aligned} &P(G_{d,k}(i) = 1) = \mathbb{E}[(G_{d,k}(i))^q] = \alpha_p = (q_g)^{i-1} p_g \\ &P(G_{d,k}(i) = 0) = \mathbb{E}[(1-G_{d,k}(i))^q] = 1-(q_g)^{i-1} p_g \\ &\mathbb{E}\left[(G_{d,k}(i) - \alpha_p)^2\right] = (q_g)^{i-1} p_g (1-(q_g)^{i-1} p_g), \end{aligned} \tag{3.5}$$

where $q_g = (1-p_g)$.

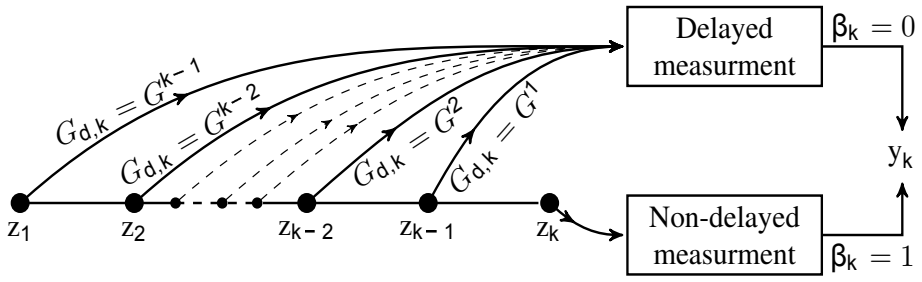


Figure 3.1: Delayed measurement representation in terms of present and past measurements.

It should be noted that the traditional Gaussian filtering method is designed for a non-delayed state-space model represented by Eqs. (3.1) and (3.2). Although a modified state-space model

for nonlinear systems with delayed measurements can be represented by Eqs. (3.1) and (3.3). Therefore, the objective of this chapter is to redesign the traditional Gaussian filtering method for the modified measurement model represented by Eqs. (3.1) and (3.3).

3.3 Gaussian Filtering with Delayed Measurements

From the detailed discussion provided in the Chapter 1, the objective of Gaussian filtering is to determine $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$ for the prediction step and $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ for the update step. In the subsequent subsections, we derive the mathematical aspects of the proposed method for performing the prediction and update steps considering that the received measurement $\mathbf{y}_k$ is delayed.

3.3.1 Prediction

This step determines $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$ using state dynamics represented by Eq. (3.1), which is independent of the modified measurement model (3.3). Therefore, this step remains similar to the ordinary Gaussian filtering approach. Thus, as $\boldsymbol{\Upsilon} = \{\boldsymbol{\Upsilon}_1, \boldsymbol{\Upsilon}_2, \cdots, \boldsymbol{\Upsilon}_{N_s}\}$ and $\mathbf{W} = \{W_1, W_2, \cdots, W_{N_s}\}$ represent the sets of deterministic sample points and associated weights, we get [23, 25]

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=1}^{N_s} W_i \boldsymbol{\Upsilon}^{\phi}_{i,k-1|k-1},$$

$$\mathbf{P}_{k|k-1} = \sum_{i=1}^{N_s} W_i (\boldsymbol{\Upsilon}^{\phi}_{i,k-1|k-1} - \hat{\mathbf{x}}_{k|k-1})(\boldsymbol{\Upsilon}^{\phi}_{i,k-1|k-1} - \hat{\mathbf{x}}_{k|k-1})^T + \mathbf{Q}_k,$$

where

$$\boldsymbol{\Upsilon}^{\phi}_{i,k-1|k-1} = \phi_{k-1}(\Sigma_{k-1|k-1}\boldsymbol{\Upsilon}_i + \hat{\mathbf{x}}_{k-1|k-1}),$$

$$\mathbf{P}_{k-1|k-1} = \Sigma_{k-1|k-1}\Sigma^T_{k-1|k-1}.$$

Please note that N_s is the number of sample points and $\boldsymbol{\Sigma}_{k-1|k-1}$ is the Cholesky decomposition of $\mathbf{P}_{k-1|k-1}$.

3.3.2 Update

As discussed earlier, the update step determines $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ based on the information received from $\mathbf{y}_k$. It is based on the statistical parameters of $\mathbf{y}_k$, *i.e.* the estimate $\hat{\mathbf{y}}_{k|k-1}$, covariance $\mathbf{P}^{\mathbf{yy}}_{k|k-1}$,

and cross-covariance $\mathbf{P}^{\mathbf{xy}}_{k|k-1}$. Subsequently, $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ are, respectively, given as [23, 25]

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

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{P}^{\mathbf{yy}}_{k|k-1} \mathbf{K}_k^T,$$

where

$$\mathbf{K}_k = \mathbf{P}^{\mathbf{xy}}_{k|k-1} (\mathbf{P}^{\mathbf{yy}}_{k|k-1})^{-1},$$

is the Kalman gain.

In the above expressions, $\hat{\mathbf{y}}_{k|k-1}$, $\mathbf{P}^{\mathbf{yy}}_{k|k-1}$, and $\mathbf{P}^{\mathbf{xy}}_{k|k-1}$ are computed for the delayed measurement $\mathbf{y}_k$ represented by the modified measurement model (3.3). As shown in Eq. (3.3), $\mathbf{y}_k$ is stochastic summation of several non-delayed measurements $\mathbf{z}_{k-j}\ \forall\ j \in \{0, 1, \cdots, k-1\}$. Therefore, $\hat{\mathbf{y}}_{k|k-1}$, $\mathbf{P}^{\mathbf{yy}}_{k|k-1}$, and $\mathbf{P}^{\mathbf{xy}}_{k|k-1}$ can also be expressed in terms of non-delayed measurement parameters, $\hat{\mathbf{z}}_{k-j|k-1}$, $\mathbf{P}^{\mathbf{zz}}_{k-j|k-1}$, and $\mathbf{P}^{\mathbf{xz}}_{k-j|k-1}$. The author determines $\hat{\mathbf{z}}_{k-j|k-1}$, $\mathbf{P}^{\mathbf{zz}}_{k-j|k-1}$, and $\mathbf{P}^{\mathbf{xz}}_{k-j|k-1}$ using the traditional Gaussian filtering method, giving [23, 25]

$$\hat{\mathbf{z}}_{k-j|k-1} = \sum_{i=1}^{N_s} W_i \boldsymbol{\Upsilon}^{\Psi}_{i,k-j|k-1},$$

$$\mathbf{P}^{\mathbf{zz}}_{k-j|k-1} = \sum_{i=1}^{N_s} W_i (\boldsymbol{\Upsilon}^{\Psi}_{i,k-j|k-1} - \hat{\mathbf{z}}_{k-j|k-1})(\boldsymbol{\Upsilon}^{\Psi}_{i,k-j|k-1} - \hat{\mathbf{z}}_{k-j|k-1})^T + \mathbf{R}_k,$$

$$\mathbf{P}^{\mathbf{xz}}_{k-j|k-1} = \sum_{i} W_i (\boldsymbol{\Upsilon}_{i,k-j|k-1} - \hat{\mathbf{x}}_{k-j|k-1})(\boldsymbol{\Upsilon}^{\Psi}_{i,k-j|k-1} - \hat{\mathbf{z}}_{k-j|k-1})^T,$$

where

$$\boldsymbol{\Upsilon}^{\Psi}_{i,k-j|k-1} = \Psi_{k-j}(\Sigma_{k-j|k-1} \boldsymbol{\Upsilon}_i + \hat{\mathbf{x}}_{k-j|k-1}),$$

$$\boldsymbol{\Upsilon}_{i,k-j|k-1} = \Sigma_{k-j|k-1} \boldsymbol{\Upsilon}_i + \hat{\mathbf{x}}_{k-j|k-1},$$

with $\mathbf{P}_{k-j|k-1} = \boldsymbol{\Sigma}_{k-j|k-1} \boldsymbol{\Sigma}^T_{k-j|k-1}$.

In the following discussion, the author computes the expressions of $\hat{\mathbf{y}}_{k|k-1}$, $\mathbf{P}^{\mathbf{yy}}_{k|k-1}$, and $\mathbf{P}^{\mathbf{xy}}_{k|k-1}$ in terms of $\hat{\mathbf{z}}_{k-j|k-1}$, $\mathbf{P}^{\mathbf{zz}}_{k-j|k-1}$, and $\mathbf{P}^{\mathbf{xz}}_{k-j|k-1}$.

Theorem 3.3.1 *Estimate of delayed measurement $\mathbf{y}_k$, i.e. $\hat{\mathbf{y}}_{k|k-1}$, can be obtained as*

$$\hat{\boldsymbol{y}}_{k|k-1} = p_b \hat{\boldsymbol{z}}_{k|k-1} + (1 - p_b) \sum_{i=1}^{k-1} (1 - p_g)^{i-1} p_g \hat{\boldsymbol{z}}_{k-i|k-1}. \tag{3.6}$$

Proof: Estimate of $\mathbf{y}_k$ is

$$\hat{\mathbf{y}}_{k|k-1} = \mathbb{E}\left[\mathbf{y}_{k|k-1}\right] = \mathbb{E}\left[\beta_k \mathbf{z}_{k|k-1} + (1-\beta_k)\sum_{i=1}^{k-1} G_{d,k}(i)\mathbf{z}_{k-i|k-1}\right].$$

Note that β_k represents the delay possibility, which is independent of $\mathbf{z}_k$ and $\mathbf{z}_{k-i}$ representing the measurement values. Similarly, $G_{d,k}(i)$ is independent of $\mathbf{z}_{k-i}$. Therefore, the above expression can be simplified as

$$\hat{\mathbf{y}}_{k|k-1} = \mathbb{E}[\beta_k]\,\mathbb{E}\left[\mathbf{z}_{k|k-1}\right] + \mathbb{E}[(1-\beta_k)]\sum_{i=1}^{k-1}\mathbf{E}[G_{d,k}(i)]\,\mathbf{E}\left[\mathbf{z}_{k-i|k-1}\right].$$

Further, $\hat{\mathbf{y}}_{k|k-1}$ is expressed as

$$\hat{\mathbf{y}}_{k|k-1} = \mathbb{E}[\beta_k]\,\hat{\mathbf{z}}_{k|k-1} + \mathbb{E}[(1-\beta_k)]\sum_{i=1}^{k-1}\mathbb{E}[G_{d,k}(i)]\,\hat{\mathbf{z}}_{k-i|k-1}.$$

Substituting $\mathbf{E}[\beta_k]$, $\mathbf{E}[(1-\beta_k)]$ and $\mathbf{E}[G_{d,k}(i)]$ from Eqs. (3.4) and (3.5), the above equation reduces to Eq. (3.6). □

Theorem 3.3.2 *The innovation covariance $\boldsymbol{P}^{\mathbf{yy}}_{k|k-1}$ can be obtained as*

$$\begin{aligned}\mathbf{P}^{\mathbf{yy}}_{k|k-1} =& p_b\mathbf{P}^{\mathbf{zz}}_{k|k-1} + p_b(1-p_b)\hat{\mathbf{z}}_{k|k-1}(\hat{\mathbf{z}}_{k|k-1})^T + (1-p_b)\sum_{i=1}^{k-1}(1-p_g)^{i-1}p_g \\ &\times \mathbf{P}^{\mathbf{zz}}_{k-i|k-1} + \sum_{i=j=1}^{k-1}\left((1-p_g)^{i-1}p_g - (1-p_b)(1-p_g)^{2i-2}p_g^2(3p_b-1)\right) \\ &\times \hat{\mathbf{z}}_{k-i|k-1}\hat{\mathbf{z}}^T_{k-i|k-1} + \sum_{i\neq j=1}^{k-1}\left[p_b(1-p_g)^{i+j-2}p_g^2 - 2p_b(1-p_b)(1-p_g)^{i+j-2}\right. \\ &\left.\times p_g^2 + (1-p_b)^2(1-p_g)^{i+j-2}p_g^2\right]\hat{\mathbf{z}}_{k-i|k-1}\hat{\mathbf{z}}^T_{k-j|k-1}.\end{aligned} \tag{3.7}$$

Proof: $\mathbf{P}^{\mathbf{yy}}_{k|k-1}$ is given as

$$\mathbf{P}^{\mathbf{yy}}_{k|k-1} = \mathbb{E}\left[(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1})(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1})^T\right]. \tag{3.8}$$

From Eqs. (3.3) and (3.6), we get

$$\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1} = \beta_k \mathbf{z}_k + (1-\beta_k)\sum_{i=1}^{k-1} G_{d,k}(i)\mathbf{z}_{k-i} - \left(p_b \hat{\mathbf{z}}_{k|k-1} + (1-p_b)\sum_{i=1}^{k-1}(1-p_g)^{i-1} p_g \hat{\mathbf{z}}_{k-i|k-1} \right).$$

It can be expressed as

$$\begin{aligned} \mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1} = & \underbrace{\beta_k(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})}_{M_1} + \underbrace{(\beta_k - p_b)\hat{\mathbf{z}}_{k|k-1}}_{M_2} + \underbrace{(1-\beta_k)\sum_{i=1}^{k-1} G_{d,k}(i)(\mathbf{z}_{k-i} - \hat{\mathbf{z}}_{k-i|k-1})}_{M_3} \\ & + \underbrace{\sum_{i=1}^{k-1}\left((1-\beta_k)G_{d,k}(i) - (1-p_b)(1-p_g)^{i-1}p_g\right)\hat{\mathbf{z}}_{k-i|k-1}}_{M_4}. \end{aligned} \tag{3.9}$$

Substituting $\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}$ from the above equation into (3.8), we obtain

$$\mathbf{P}^{\mathbf{yy}}_{k|k-1} = \sum_{i=1}^{4}\sum_{j=1}^{4} \mathbb{E}\left[M_i M_j^T\right].$$

Using the joint estimation property of random variables, we can show that $\mathbb{E}[M_i M_j^T] = 0 \; \forall \; i \neq j$. Thus,

$$\mathbf{P}^{\mathbf{yy}}_{k|k-1} = \sum_{i=1}^{4} \mathbb{E}\left[M_i M_i^T\right]. \tag{3.10}$$

For the M_1 expression given in Eq. (3.9), we can write

$$\mathbb{E}\left[M_1 M_1^T\right] = \mathbb{E}\left[\beta_k^2(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})^T\right].$$

Note that β_k is independent of $\mathbf{z}_k$ and $\hat{\mathbf{z}}_{k|k-1}$ is a constant. Therefore, substituting $\mathbb{E}\left[\beta_k^2\right]$ from Eq. (3.4), we get

$$\mathbb{E}\left[M_1 M_1^T\right] = p_b P^{\mathbf{zz}}_{k|k-1}. \tag{3.11}$$

$\mathbb{E}\left[M_2 M_2^T\right]$ with M_2 expression given in Eq. (3.9) can be written as

$$\mathbb{E}\left[M_2 M_2^T\right] = \mathbb{E}\left[(\beta_k - p_b)^2 \hat{\mathbf{z}}_{k|k-1}\hat{\mathbf{z}}_{k|k-1}^T\right].$$

Then, we get

$$\mathbb{E}\left[M_2 M_2^T\right] = \mathbb{E}\left[(\beta_k - p_b)^2\right] \mathbb{E}\left[\hat{\mathbf{z}}_{k|k-1} \hat{\mathbf{z}}_{k|k-1}^T\right].$$

Substituting $\mathbb{E}\left[(\beta_k - p_b)^2\right]$ from Eq. (3.4), we get

$$\mathbb{E}\left[M_2 M_2^T\right] = p_b(1-p_b^2)\hat{\mathbf{z}}_{k|k-1} \hat{\mathbf{z}}_{k|k-1}^T. \tag{3.12}$$

For the M_3 term given in Eq. (3.9), we can write

$$\mathbb{E}\left[M_3 M_3^T\right] = \mathbb{E}\Big[(1-\beta_k) \sum_{i=1}^{k-1} G_{d,k}(i)(\mathbf{z}_{k-i} - \hat{\mathbf{z}}_{k-i|k-1})(1-\beta_k) \sum_{j=1}^{k-1} G_{d,k}(j)(\mathbf{z}_{k-j} - \hat{\mathbf{z}}_{k-j|k-1})^T\Big]$$

Using the independent property of the random variables, it can be shown that all the terms for $i \neq j$ are zero. Therefore, $\mathbb{E}\left[M_3 M_3^T\right]$ can be simplified as

$$\mathbb{E}\left[M_3 M_3^T\right] = \mathbb{E}\left[(1-\beta_k)^2\right] \sum_{i=1}^{k-1} \Big(\mathbb{E}\left[G_{d,k}(i)^2\right] \mathbb{E}\left[(\mathbf{z}_{k-i} - \hat{\mathbf{z}}_{k-i|k-1})(\mathbf{z}_{k-i} - \hat{\mathbf{z}}_{k-i|k-1})^T\right]\Big).$$

Substituting $\mathbb{E}\left[(1-\beta_k)^2\right]$ and $\mathbb{E}\left[G_{d,k}(i)^2\right]$ from Eqs. (3.4) and (3.5), respectively, the above expression can be simplified as

$$\mathbb{E}\left[M_3 M_3^T\right] = (1-p_b) \sum_{i=j=1}^{k-1} (1-p_g)^{i-1} p_g \mathbf{P}_{k-i|k-1}^{\mathbf{zz}}. \tag{3.13}$$

Lastly, $\mathbb{E}\left[M_4 M_4^T\right]$ with M_4 given in Eq. (3.9) can be expressed in the form of Eq. (3.14). A

$$\begin{aligned}\mathbb{E}\big[M_4 M_4^T\big] =& \mathbb{E}\big[\sum_{i=1}^{k-1} \big((1-\beta_k)G_{d,k}(i) - (1-p_b)(1-p_g)^{i-1}p_g\big)\hat{\mathbf{z}}_{k-i|k-1} \sum_{j=1}^{k-1} \big((1-\beta_k)G_{d,k}(j) \\ & - (1-p_b)(1-p_g)^{j-1}p_g\big)\hat{\mathbf{z}}_{k-j|k-1}^T\big].\end{aligned} \tag{3.14}$$

further simplification of Eq. (3.14) leads to Eq. (3.15) and is given in Eq.(3.16).

Substituting $\mathbb{E}[M_1 M_1^T]$, $\mathbb{E}[M_2 M_2^T]$, $\mathbb{E}[M_3 M_3^T]$ and $\mathbb{E}[M_4 M_4^T]$ from Eq. (3.11), (3.12), (3.13) and (3.16), respectively, into Eq. (3.8), $\mathbf{P}_{k|k-1}^{\mathbf{yy}}$ can be simplified in the form of Eq. (3.7). □

$$\mathbb{E}\left[M_4 M_4^T\right] = \sum_{i=j=1}^{k-1} \mathbb{E}\left[\left((1-\beta_k)G_{d,k}(i) - (1-p_b)(1-p_g)^{i-1}p_g\right)^2\right] \mathbb{E}\left[\hat{\mathbf{z}}_{k-i|k-1}\hat{\mathbf{z}}_{k-i|k-1}^T\right] + \sum_{i\neq j=1}^{k-1} \mathbb{E}[((1-\beta_k)G_{d,k}(i) - (1-p_b)(1-p_g)^{i-1}p_g)((1-\beta_k)G_{d,k}(j) - (1-p_b)(1-p_g)^{j-1}p_g)]\mathbb{E}\left[\hat{\mathbf{z}}_{k-j|k-1}\hat{\mathbf{z}}_{k-j|k-1}^T\right]. \tag{3.15}$$

$$\mathbb{E}\left[M_4 M_4^T\right] = \sum_{i=j=1}^{k-1} (p_b(1-p_g)^{i-1}p_g - (1-p_b)(1-p_g)^{2i-2}p_g^2(3p_b-1))\hat{\mathbf{z}}_{k-i|k-1}\hat{\mathbf{z}}_{k-i|k-1}^T + \sum_{i\neq j=1}^{k-1} \Big(p_b(1-p_g)^{i+j-2}p_g^2 - 2p_b(1-p_b)(1-p_g)^{i+j-2}p_g^2 + (1-p_b)^2 \times (1-p_g)^{i+j-2}p_g^2\Big)\hat{\mathbf{z}}_{k-i|k-1}\hat{\mathbf{z}}_{k-j|k-1}^T. \tag{3.16}$$

Theorem 3.3.3 *The cross-covariance matrix between* $\boldsymbol{x}$ *and* $\boldsymbol{y}$ *at* k^{th} *time instant can be given as*

$$\boldsymbol{P}_{k|k-1}^{\boldsymbol{xy}} = p_b \boldsymbol{P}_{k|k-1}^{\boldsymbol{xz}} + p_b \sum_{i=1}^{k-1} (1-p_g)^{i-1} p_g \boldsymbol{P}_{k-i|k-1}^{\boldsymbol{xz}}. \tag{3.17}$$

Proof: The cross-covariance matrix between $\mathbf{x}$ and $\mathbf{y}$ at the k^{th} instant is

$$\mathbf{P}_{k|k-1}^{\mathbf{xy}} = \mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1})^T\right].$$

Substituting $\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1} = \sum_{i=1}^{4} M_i$ from Eq. (3.9),

$$\mathbf{P}_{k|k-1}^{\mathbf{xy}} = \sum_{i=1}^{4} \mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})M_i^T\right]. \tag{3.18}$$

Applying joint estimation property of the independent random variables, it can be shown that

$$\mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})M_2^T\right] = \mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})M_4^T\right] = 0. \tag{3.19}$$

Therefore, we provide the computational aspects for the non-zero terms, $\mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})M_1^T\right]$ and $\mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})M_3^T\right]$, only.

The term corresponding to M_1 can be obtained as

$$\mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})M_1^T\right] = \mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})\beta_k(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})^T\right].$$

Then, we get

$$\mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})M_1^T\right] = p_b P^{\mathbf{xz}}_{k|k-1}. \tag{3.20}$$

Similarly, the term corresponding to M_3 can be obtained as

$$\mathbb{E}\left[\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}\right)M_3^T\right] = \mathbb{E}\left[\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}\right)\left((1-\beta_k)\sum_{i=1}^{k-1} G_{d,k}(i)(\mathbf{z}_{k-i} - \hat{\mathbf{z}}_{k-i|k-1})^T\right)\right].$$

It can be further simplified as

$$\mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})M_3^T\right] = \mathbb{E}\left[(1-\beta_k)\right]\sum_{i=1}^{k-1}\left(\mathbb{E}\left[G_{d,k}(i)\right]\mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{z}_{k-i} - \hat{\mathbf{z}}_{k-i|k-1})^T\right]\right).$$

Substituting $\mathbb{E}\left[(1-\beta_k)\right]$ and $\mathbb{E}\left[G_{d,k}(i)\right]$ from Eqs. (3.4) and (3.5), respectively, we get

$$\mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})M_3^T\right] = \sum_{i=1}^{k-1}(1-p_b)(1-p_g)^{i-1}p_g\mathbf{P}^{\mathbf{xz}}_{k-i|k-1}. \tag{3.21}$$

Substituting various terms of Eq. (3.18) from Eqs. (3.19), (3.20), and (3.21), we get $\mathbf{P}^{\mathbf{xy}}_{k|k-1}$ in the form of Eq. (3.17). □

As discussed earlier, the traditional Gaussian filtering algorithm is based on the statistical measures of $\mathbf{z}_k$, *i.e.* $\hat{\mathbf{z}}_{k|k-1}$, $\mathbf{P}^{\mathbf{zz}}_{k|k-1}$, and $\mathbf{P}^{\mathbf{xz}}_{k|k-1}$, which ignores the delay. It can be modified for delayed measurements by replacing $\hat{\mathbf{z}}_{k|k-1}$, $\mathbf{P}^{\mathbf{zz}}_{k|k-1}$, and $\mathbf{P}^{\mathbf{xz}}_{k|k-1}$ with $\hat{\mathbf{y}}_{k|k-1}$, $\mathbf{P}^{\mathbf{yy}}_{k|k-1}$, and $\mathbf{P}^{\mathbf{xy}}_{k|k-1}$, respectively. A block diagram representing the steps to be followed for implementing the proposed filtering algorithm is presented in Fig. 3.2. As the statistical measures $\hat{\mathbf{y}}_{k|k-1}$, $\mathbf{P}^{\mathbf{yy}}_{k|k-1}$, and $\mathbf{P}^{\mathbf{xy}}_{k|k-1}$ account for delay, the proposed filtering method is expected to outperform the traditional Gaussian filtering in the presence of delay.

The existing delay filters [97], [99] assign one Bernoulli random variable for each possible delay extent in their delay modeling strategies. For example, if the maximum delay is restricted to four time-steps (*i.e.*, up to four sampling intervals), they assign five Bernoulli random variables,

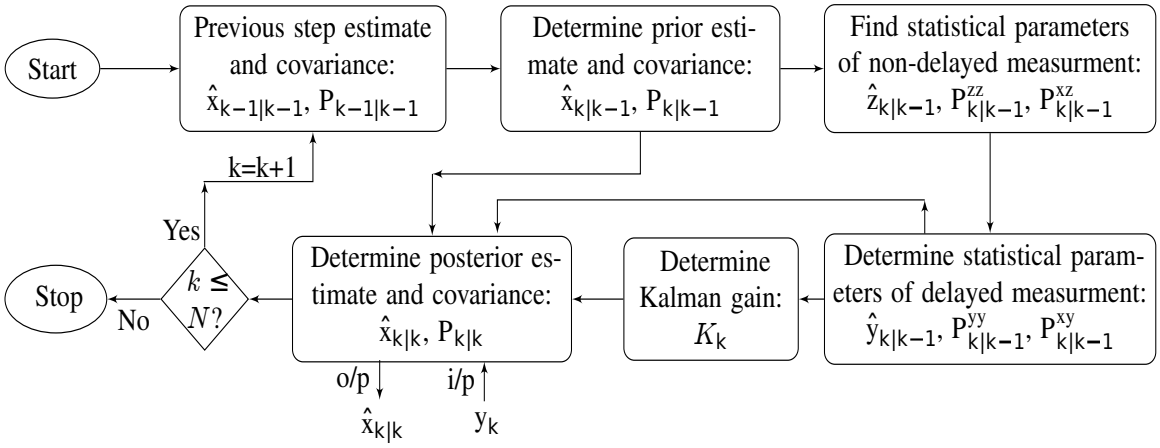


Figure 3.2: Block diagram representing the steps for implementing the proposed filtering via computing the statistical measures $\hat{\mathbf{y}}_{k|k-1}$, $\mathbf{P}^{\mathbf{yy}}_{k|k-1}$, and $\mathbf{P}^{\mathbf{xy}}_{k|k-1}$ with stored statistical information using $\hat{\mathbf{z}}_{k|k-1}$, $\mathbf{P}^{\mathbf{zz}}_{k|k-1}$, and $\mathbf{P}^{\mathbf{xz}}_{k|k-1}$.

one for the non-delayed case and four for the four past sampling instants within the range of maximum delay. At any instant t_k, only one of all Bernoulli random variables can have a unity value, while others necessarily remain zero. The measurement corresponds to the time instant for which the Bernoulli random variable has the unity value. Alternatively, the unity value of Bernoulli random variables determines whether the measurement belongs to the current instant (non-delayed measurement) or a particular past instant (delayed measurement).

Furthermore, the existing delay filters [97], [99] assume that the probability of each Bernoulli random variable being one is known, which also implies the probability of the corresponding delay. Thus, it can be stated that they require the probability of every possible delay to be known for the filtering. Please note that these probabilities are mostly unknown or inaccurately known in practice, resulting in poor accuracy. On the other hand, the proposed method requires only two probabilities to characterize any large delay scenario: one for characterizing no-delay case and other for characterizing any large delay. Furthermore, the existing methods [97], [93],[99] require to generate one random number for each possible delay extent for the measurement data simulation. Considering that the delay extent may be up to multiple sampling intervals, they require generating multiple Bernoulli random numbers with zero or one outcomes. As only one measurement is received at any time, they assume that only one of these Bernoulli random variables can have an outcome 'one', and the rest must remain 'zero'. At the same time, the independent assumption of all Bernoulli random variables contradicts this assumption, leading to ambiguity in

data simulation. On the other hand, the proposed method precludes such ambiguities by directly generating a sequence of zero and one from the information of only one delay probability.

The proposed method utilizes information from every past sampling instants, as conclusive from Eqs. (3.6), (3.7), and (3.17). Therefore, its computational demand may excessively increase in long-time processing. However, the contributions from the excessive past instants become negligible, as $q_g < 1$. Thus, the practitioners may fix threshold weights for excluding the excessive past instants. Subsequently, the practitioners can restrict the computational demand within a permissible limit without damaging the accuracy considerably. Nevertheless, this restriction applies only for a long-time processing of the proposed method with a limited computational budget. Thus, in general, the proposed method remains superior to the existing methods [97, 99] handling finite delays with possibilities of missing measurements.

Remark 3.3.1 *Any of the existing Gaussian filters, like EKF, UKF, CKF, and GHF, can be extended with the proposed filtering technique. Therefore, the various filtering alternatives available to achieve a trade-off between the accuracy level and the computational budget are applicable to the proposed filtering technique as well.*

Remark 3.3.2 *The increased computational time of the proposed filtering technique is not a major concern if the sampling interval is not very small, i.e. the filter gets enough processing time before a new measurement is available.*

Remark 3.3.3 *Although the storage requirement of the proposed filtering technique is relatively high, it is not arbitrarily large. Therefore, it is expected to cause only a minor increase in the storage budget.*

3.3.3 Multiple sensor environment

The above discussed filtering algorithm for delayed measurements assumes an equal delay probability for each measurement element. However, the practical measurements are often composed of various elements received from different sensing units. Please note that each sensor may have a different delay, leading to different delays for different elements of a measurement.

If n_r is the number of sensors, then $\mathbf{z}_k = [\mathbf{z}_k^1, \mathbf{z}_k^2, \cdots, \mathbf{z}_k^{n_r}]$ $\left(\mathbf{y}_k = [\mathbf{y}_k^1, \mathbf{y}_k^2, \cdots, \mathbf{y}_k^{n_r}]\right)$, where $\mathbf{z}_k^j$ $(\mathbf{y}_k^j)$ represents the element of $\mathbf{z}_k$ $(\mathbf{y}_k)$ received from the j^{th} sensor. Let us denote β_k^j and $G_{d_j,k}$ as the

Bernoulli and geometric random variables, respectively associated with $\mathbf{y}_k^j$, then

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{y}_k^1 \\ \mathbf{y}_k^2 \\ \vdots \\ \mathbf{y}_k^{n_r} \end{bmatrix} = \begin{bmatrix} \beta_k^1 \mathbf{z}_k^1 + (1-\beta_k^1)\sum_{i=1}^{d_1} G_{d_1,k}(i)\mathbf{z}_{k-i}^1 \\ \beta_k^2 \mathbf{z}_k^2 + (1-\beta_k^2)\sum_{i=1}^{d_2} G_{d_2,k}(i)\mathbf{z}_{k-i}^2 \\ \vdots \\ \beta_k^r \mathbf{z}_k^r + (1-\beta_k^r)\sum_{i=1}^{d_r} G_{d_r,k}(i)\mathbf{z}_{k-i}^{n_r} \end{bmatrix},$$

where d_j is the delay in the j^{th} sensor data at the k^{th} instant.

Following the steps of Theorem 3.3.1 for each element of $\mathbf{y}_k$, $\hat{\mathbf{y}}_{k|k-1}$ can be derived as

$$\hat{\mathbf{y}}_{k|k-1} = \begin{bmatrix} p_b^1 \hat{\mathbf{z}}_{k|k-1}^1 + (1-p_b^1)\sum_{i=1}^{k-1}\left(1-p_g^1\right)^{i-1} p_g^1 \hat{\mathbf{z}}_{k-i|k-1}^1 \\ p_b^2 \hat{\mathbf{z}}_{k|k-1}^2 + (1-p_b^2)\sum_{i=1}^{k-1}\left(1-p_g^2\right)^{i-1} p_g^2 \hat{\mathbf{z}}_{k-i|k-1}^2 \\ \vdots \\ p_b^{n_r} \hat{\mathbf{z}}_{k|k-1}^{n_r} + (1-p_b^{n_r})\sum_{i=1}^{k-1}\left(1-p_g^{n_r}\right)^{i-1} p_g^{n_r} \hat{\mathbf{z}}_{k-i|k-1}^{n_r} \end{bmatrix}.$$

The above expression shows that

$$\hat{\mathbf{y}}_{k|k-1} = \left[\hat{\mathbf{y}}_{k|k-1}^1, \hat{\mathbf{y}}_{k|k-1}^2, \cdots, \hat{\mathbf{y}}_{k|k-1}^{n_r}\right]^T. \tag{3.22}$$

Eq. (3.22) shows that each element of $\mathbf{y}_k$ can be considered independently when filtering with multi-sensor data. Similarly, $\mathbf{P}_{k|k-1}^{\mathbf{yy}}$ and $\mathbf{P}_{k|k-1}^{\mathbf{xy}}$ can also be obtained for the individual sensors. Afterward, they can be combined together to deal with multi-sensor data.

3.4 Simulation Results

This section implements the proposed method for two nonlinear filtering problems with delay possibilities. In both problems, the modified measurement model presented in Eq. (3.3) is used only for generating the simulated data of delayed measurements in lack of real delay data. The implementation is performed in Matlab 2019 over a personal computer with the configuration mentioned in Section 1.6. The performance is analyzed for the CKF-based formulation of Gaussian filtering, as the CKF offers high accuracy at a very low computation demand. The CKF under the proposed modification is abbreviated as CKF_GRD, while the CKF under the existing delay filtering approaches is abbreviated as CKF_RD [97] and MLCKF [99]. The CKF_RD and MLCKF

are implemented by fixing an upper-bound of delay as four sampling intervals. It should be noted that the other existing delay filters [92, 95, 93] are excluded from the benchmark comparisons because they are designed for small delays (up to one or two sampling intervals), and fail to handle higher delays.

As discussed in the Section 1.6, we choose RMSE as the performance metric, which can be defined at t_k as

$$\text{RMSE}_k = \sqrt{\frac{1}{T_{mc}} \sum_{i=1}^{T_{mc}} \left(\mathbf{x}_k^i - \hat{\mathbf{x}}_{k|k}\right)^2} \tag{3.23}$$

where $\mathbf{x}_k^i$ and $\hat{\mathbf{x}}_{k|k}^i$ are the true and the estimated states at t_k time-step and in i^{th} simulation run, and T_{mc} is the number of Monte-carlo simulations. We also obtain average RMSE (ARMSE) as

$$\text{ARMSE} = \frac{1}{T_s} \sum_{k=1}^{T_s} \text{RMSE}_k \tag{3.24}$$

where T_s is the number of time steps.

3.4.1 Problem 1

This problem is a univariate non-stationary growth model, which is widely used to validate the filtering performance [95] due to its highly nonlinear and bimodal nature. For this problem, the dynamic state space model is given as

$$\mathbf{x}_k = a_m \mathbf{x}_{k-1} + b_m \frac{\mathbf{x}_{k-1}}{1 + \mathbf{x}_{k-1}^2} + c_m \cos(1.2k\tau) + \nu_k, \tag{3.25}$$

$$\mathbf{z}_k = \mathbf{x}_k^2 / d_m + \eta_k. \tag{3.26}$$

The simulated data of the true states and measurements are generated by considering $a_m = 5$, $b_m = 25$, $c_m = 8$, $d_m = 20$, $\tau = 1$, $\mathbf{x}_0 = 0.1$ and $\mathbf{Q}_k = \mathbf{R}_k = 0.2$. The initial estimate and covariance are taken as 0.5 and 1, respectively. The filters are implemented over $T_s = 200$ time-steps and 500 Monte-Carlo simulations are performed for computing ARMSE. The true state is compared with the estimated state obtained for CKF_GRD in Fig. 3.3. A close match between the true and the estimated states concludes a successful estimation for the CKF_GRD, *i.e.* for the proposed filtering method. Fig. 3.4 shows the ARMSEs plotted for varying delay probability (*i.e.*, $1 - p_b$) and varying the probability of delay extent if the delay has occurred (*i.e.*, p_g). The ARMSE plots show a reduced ARMSE for the CKF_GRD compared to the CKF_RD, MLCKF, and CKF in the

presence of delay. We further compare the mean RMSEs of the CKF_GRD with the CKF for varying p_b and p_g in Table 3.1. The table shows a reduced RMSE of the CKF_GRD even as the p_b and p_g vary.

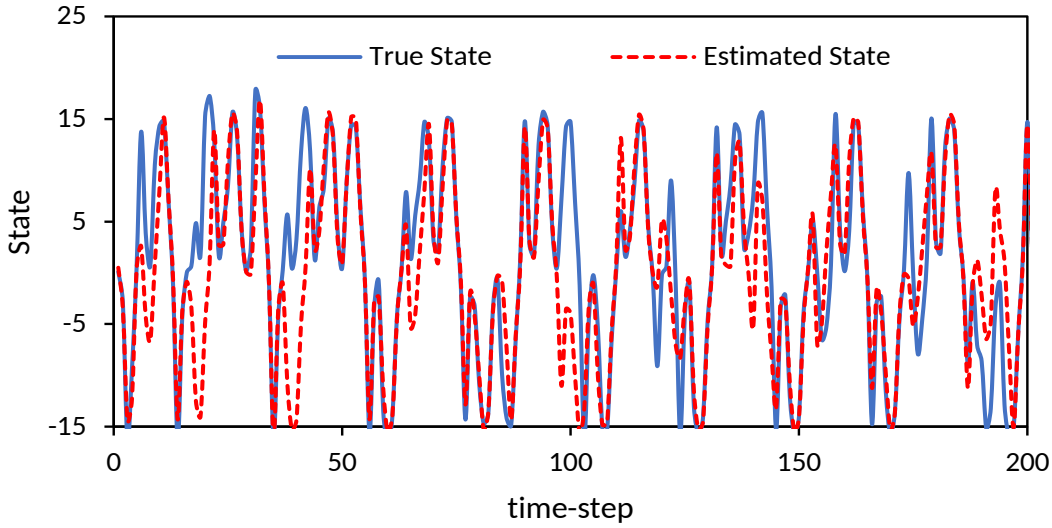


Figure 3.3: Problem1: Plots of true and estimated states.

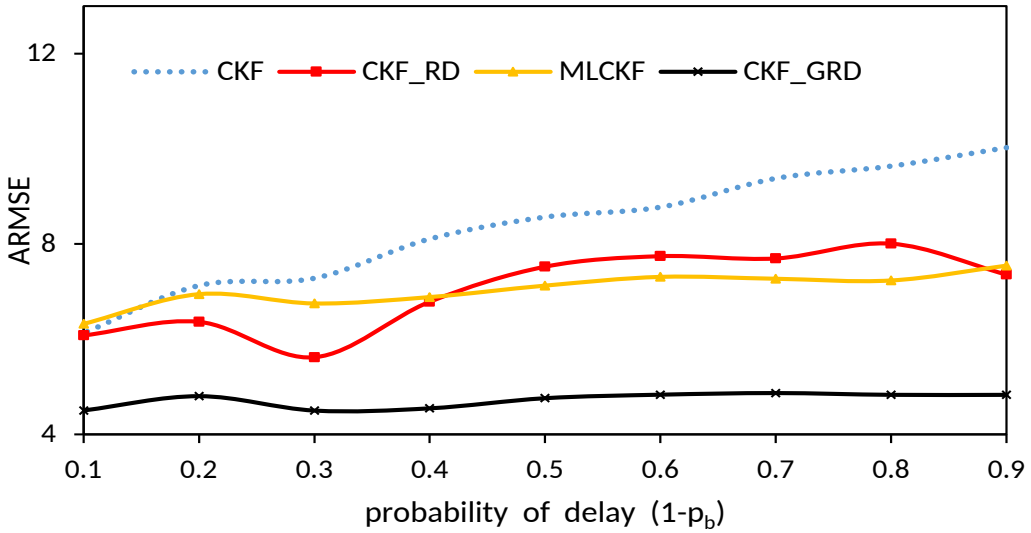
It should be mentioned that as p_b and p_g increase, the chances of delay become lesser, and even if the delay occurs, it becomes smaller. Subsequently, the performance of the CKF_GRD should get closer to the ordinary CKF as p_b and p_g increase. The same is reflected in Table 3.1.

Table 3.1: Problem 1: Average RMSE of different filters for varying p_b and p_g.

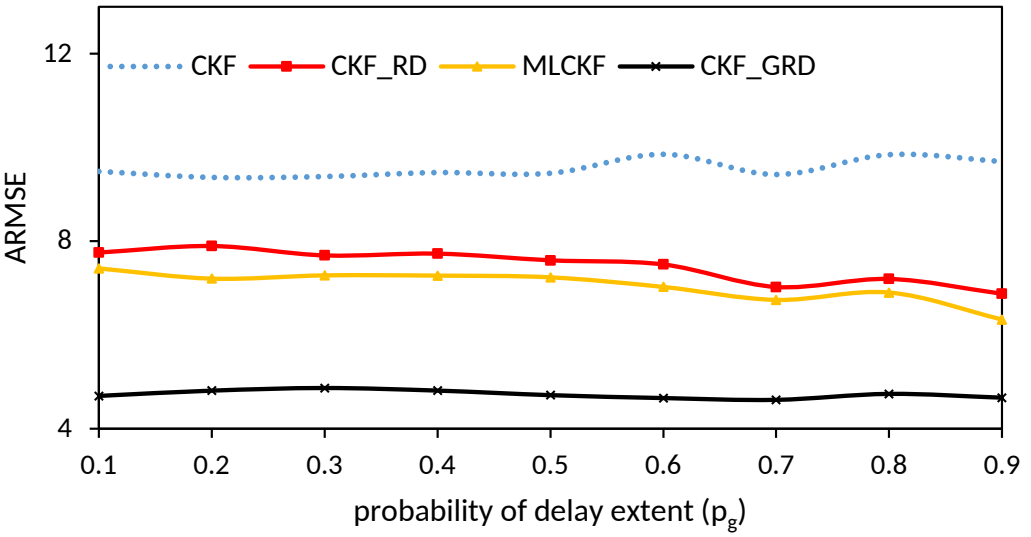
p_b	p_g	CKF	CKF_GRD
0.1	0.2	10.0912	4.7817
	0.5	10.5520	4.8928
	0.8	10.7801	4.7883
0.5	0.2	9.1307	6.3041
	0.5	9.5795	6.3209
	0.8	9.7241	6.2390
0.8	0.2	7.1977	5.1459
	0.5	7.3928	5.4609
	0.8	7.5664	5.4882

3.4.2 Problem 2

The second problem is filtering algorithm-based characterization of multiple sinusoidals superimposed in a single signal. This problem frequently appears in power [155] and communication



(a)



(b)

Figure 3.4: Problem 1: ARMSE analysis: (a) varying delay probability $(1 - p_b)$ with $p_g = 0.3$ and (b) varying probability of delay extent (p_g) with $p_b = 0.3$.

systems [156], where each sinusoidal can be characterized by frequency and amplitude. Therefore, the objective is to estimate the frequency and amplitude of each sinusoidal from the measurements available for the superimposed signal. Considering the number of sinusoidals to be three, and denoting f_i and a_i as the frequency and amplitude of the i^{th} sinusoidal, respectively, we can define $\mathbf{x} = [f_1, f_2, f_3, a_1, a_2, a_3]^T$. As the frequency and amplitude of each sinusoidal is expected to be constant, we obtain

$$\mathbf{x}_k = I_6 \mathbf{x}_{k-1} + \nu_k. \tag{3.27}$$

As the measured signal is a superimposition of the three sinusoidals to be estimated, we can define

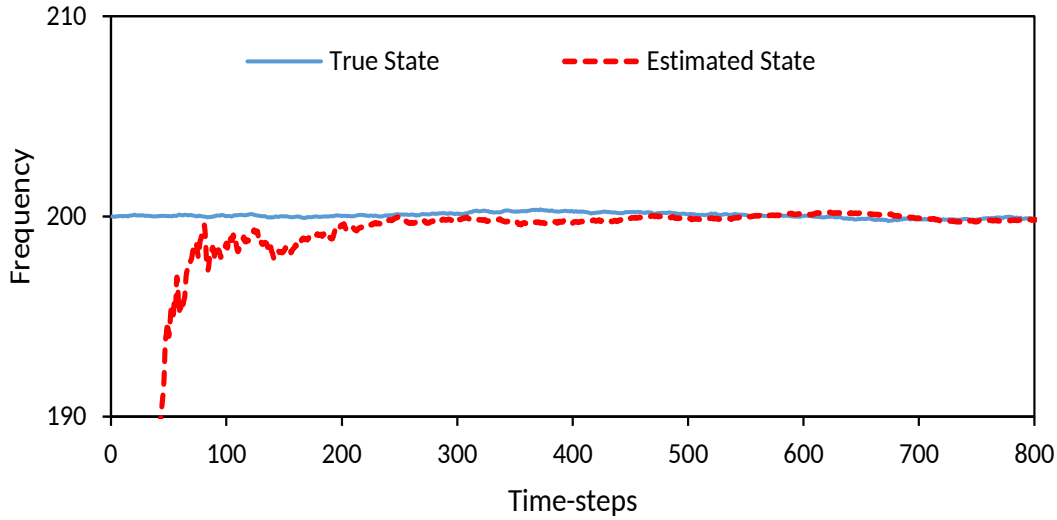
$$\mathbf{z}_k = \left[\sum_{j=1}^{3} a_{j,k} \cos(2\pi f_{j,k} k\tau) \;\; \sum_{j=1}^{3} a_{j,k} \sin(2\pi f_{j,k} k\tau) \right]^T + \eta_k, \tag{3.28}$$

where τ is the sampling time considered as $0.1667\ ms$. We consider the noise covariances as $\mathbf{Q}_k = diag([\sigma_f^2\ \sigma_f^2\ \sigma_f^2\ \sigma_a^2\ \sigma_a^2\ \sigma_a^2])$ and $\mathbf{R}_k = diag([\sigma_r^2\ \ \sigma_r^2])$, with $\sigma_f = 15\ mHz$, $\sigma_a = 20\ mV$, and $\sigma_r = 0.3\ V$. The initial true and estimated states are taken as $\mathbf{x}_0 = [200, 1000, 2200, 2, 2, 2]^T$ and $\hat{\mathbf{x}}_{0|0} = \mathcal{N}(\mathbf{x}_0, \mathbf{P}_{0|0})$, respectively, where the initial covariance is $\mathbf{P}_{0|0} = diag\,([40, 40, 40, 0.05, 0.05, 0.05])$. The filtering is performed over $T_S = 800$ time-steps and the ARMSE is computed by implementing 500 Monte-Carlo runs.

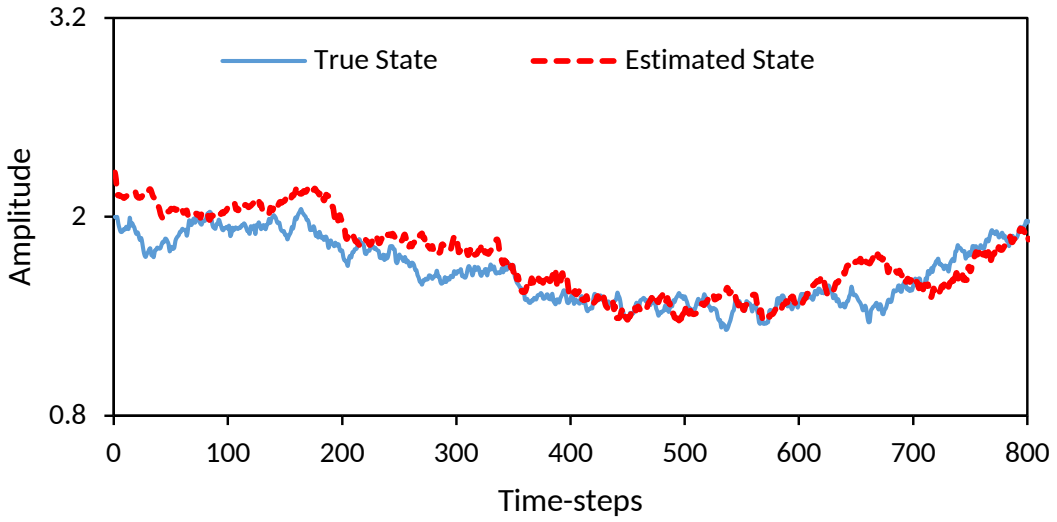
The true and estimated plots of frequency and amplitude are shown in Fig. 3.5 for the first sinusoidal. A similar pattern is observed for the other sinusoidals as well.

The ARMSE analysis for the frequency and amplitude is shown in Fig. 3.6 and 3.7, respectively. The ARMSEs shown in the figures are computed as by averaging (over the time-steps) the mean of the RMSEs obtained for the three sinusoidals. The figures show a considerably reduced ARMSE for the CKF_GRD compared to the CKF_RD, MLCKF, and ordinary CKF. On the other hand, the MLCKF and CKF_RD may perform relatively close to the CKF_GRD in some cases due to stochastic behavior of delay often matching the upper-bound closely. However, the possibility of an inappropriate upper-bound is high as there is no specific rule defined for this selection.

We further study the performance of the proposed filter in different environments that are defined by varying delay possibilities and delay extents. In this regard, we compare the average RMSE of the CKF_GRD with the CKF for varying p_b and p_g in Table 3.2 and Table 3.3. The tables conclude a reduced RMSE for the CKF_GRD even as the p_b and p_g vary. Alternatively, the tables conclude the consistently improved accuracy of the CKF_GRD when both the delay



(a)



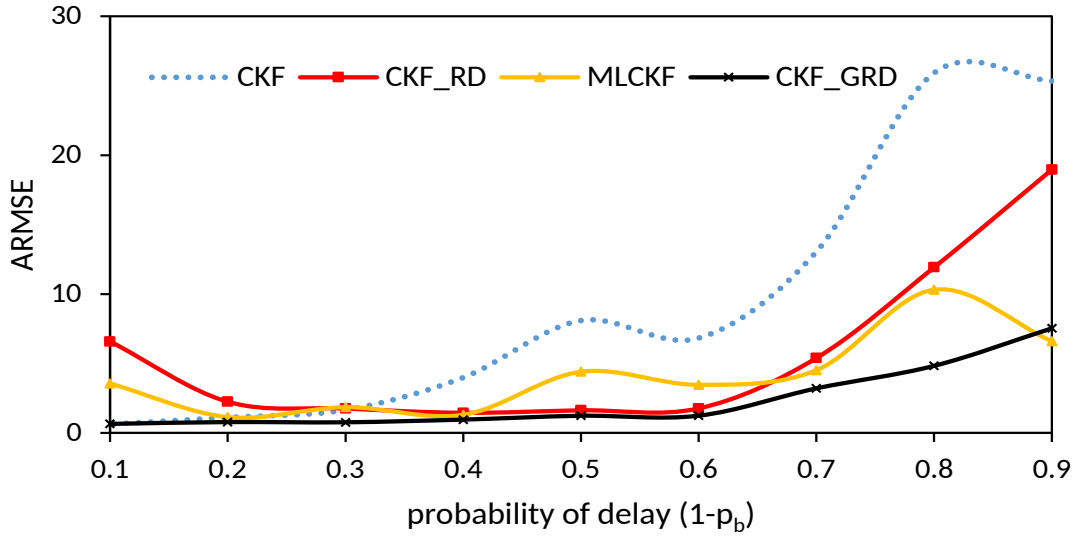
(b)

Figure 3.5: Problem 2: True and estimated plots of frequency and amplitude for the first sinusoidal.

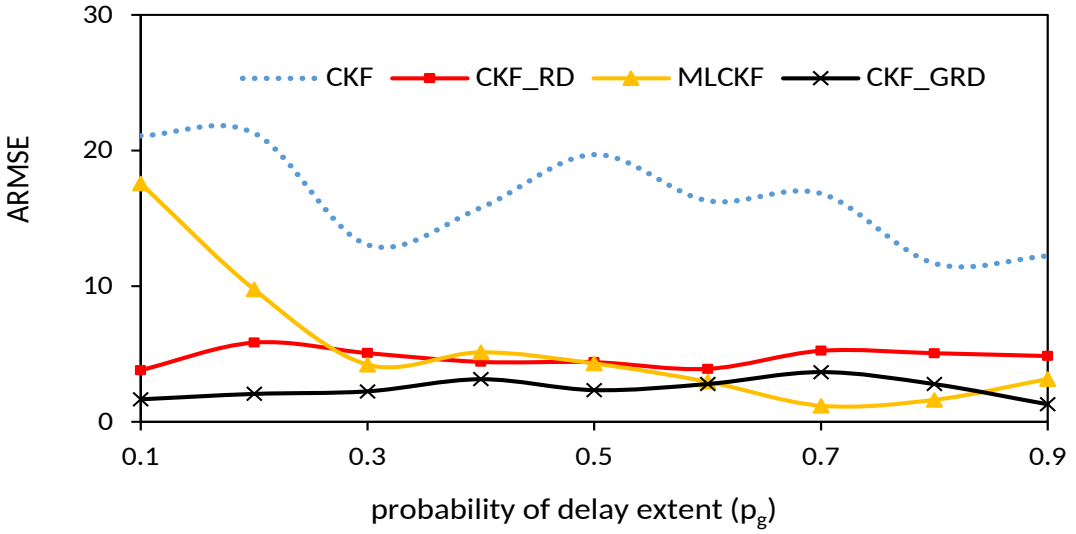
possibility and delay extents increase.

Following the discussion in the previous subsection, the performance of the CKF_GRD should become similar to the ordinary CKF as p_b and p_g become large. The same is reflected in the Table 3.2 and Table 3.3. The average RMSEs of the CKF_GRD and the CKF become very close as both p_b and p_g are assigned with a large value of 0.8.

The relative computational times for the four filters are computed with the time for CKF taken as unity, and others found as 1.1245 for CKF_RD, 1.2105 for MLCKF, and 1.4073 for CKF_GRD. It concludes that the computational time of the CKF_GRD is marginally increased.



(a)

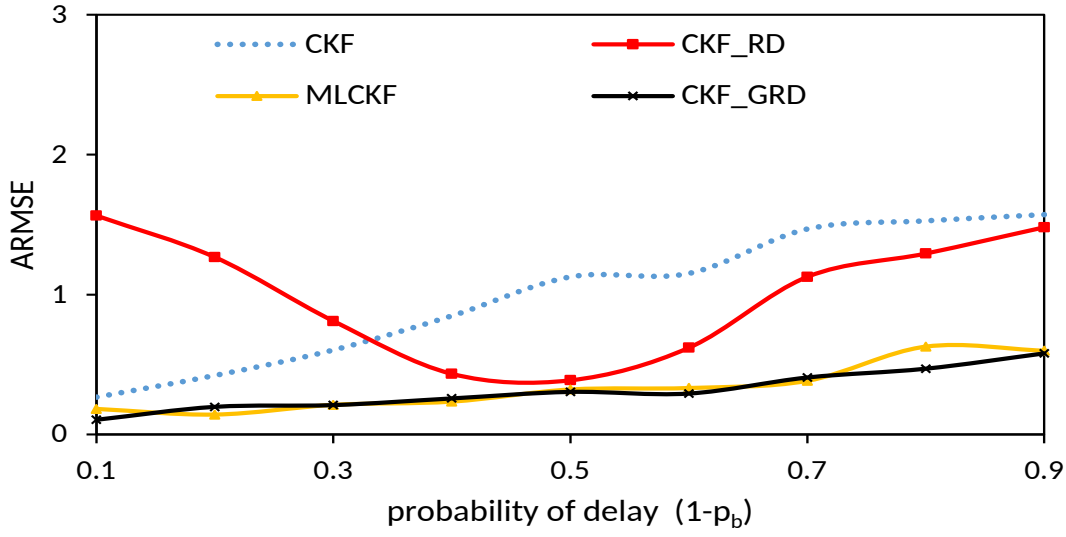


(b)

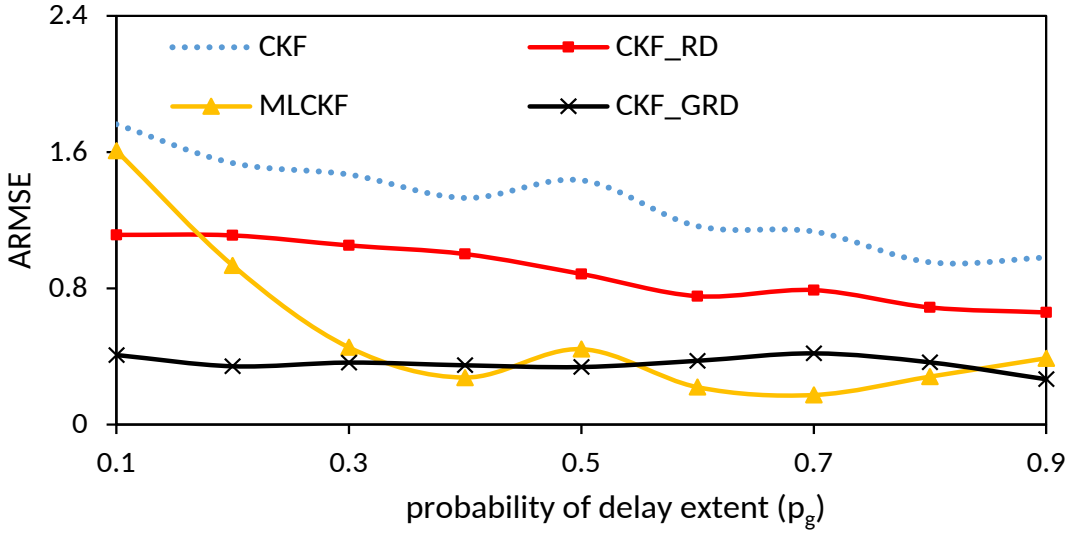
Figure 3.6: Problem 2: ARMSE analysis for frequency: (a) varying delay probability $(1-p_b)$ with $p_g = 0.3$, (b) varying probability of delay extent (p_g) with $p_b = 0.3$.

Table 3.2: Problem 2: Average RMSE of frequency obtained for different filters as p_b and p_g vary.

p_b	p_g	CKF	CKF_GRD
0.1	0.2	29.7092	7.9437
	0.5	17.8846	6.6574
	0.8	8.8956	5.6556
0.5	0.2	6.1836	1.0539
	0.5	5.0139	0.9688
	0.8	4.7383	0.8750
0.8	0.2	0.7814	0.7401
	0.5	0.8521	0.7386
	0.8	0.9737	0.7458



(a)



(b)

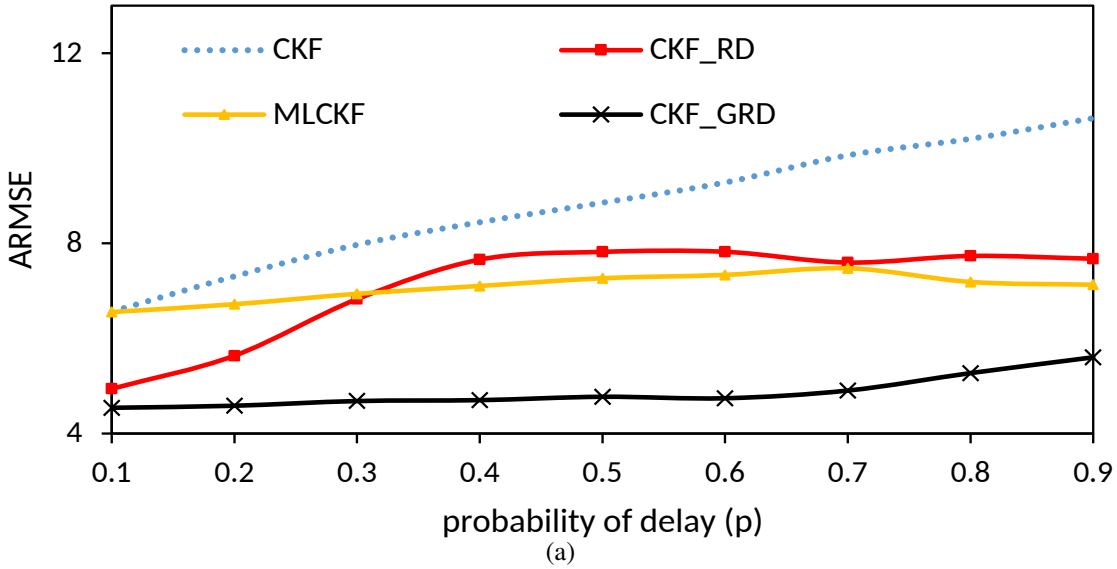
Figure 3.7: Problem 2: ARMSE analysis for amplitude: (a) varying delay probability $(1-p_b)$ with $p_g = 0.3$, (b) varying probability of delay extent (p_g) with $p_b = 0.3$.

Table 3.3: Problem 2: Average RMSE of amplitude state for different filters as p_b and p_g vary.

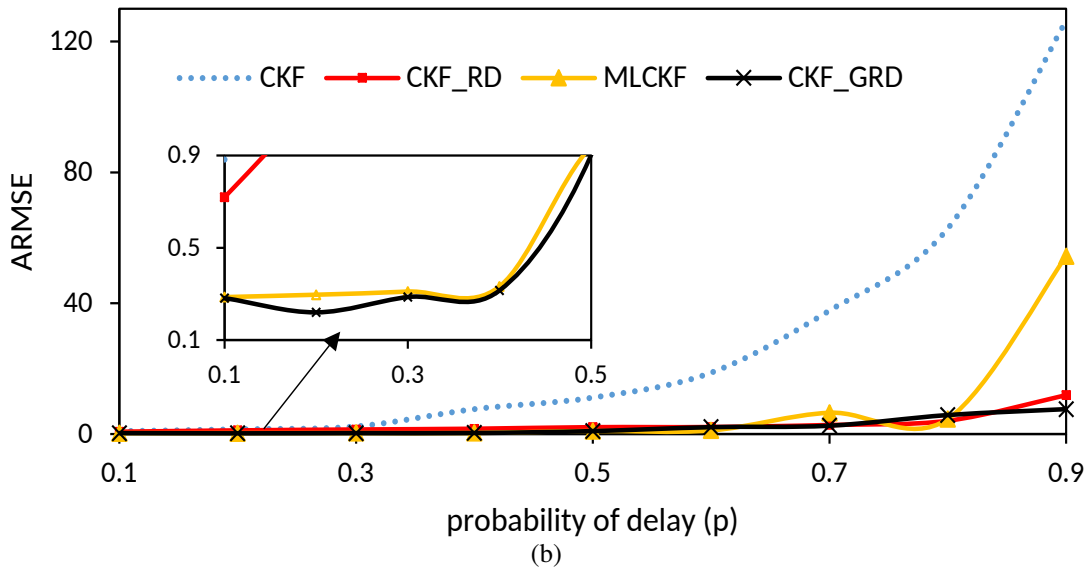
p_b	p_g	CKF	CKF_GRD
0.1	0.2	1.7046	0.6362
	0.5	1.4823	0.7370
	0.8	1.1763	0.7890
0.5	0.2	1.0977	0.2716
	0.5	0.9383	0.2419
	0.8	0.8231	0.2140
0.8	0.2	0.3939	0.2054
	0.5	0.3739	0.1955
	0.8	0.3577	0.1860

3.4.3 Performance analysis for mismatched models

We extend the performance analysis of the CKF_GRD in a new scenario, where the actual delay model is mismatched from our model. In this regard, we generate the delay following the model used in [97, 99] with delay parameter p and implement the CKF_GRD. The resulting average RMSEs of the CKF_GRD, CKF_RD, MLCKF, and CKF are shown in Fig. 3.8 for both the previously discussed problems. The figure excludes the amplitude ARMSE plot for the second problem as it is similar to the frequency. Interestingly, the ARMSE plots conclude an improved accuracy for CKF_GRD even as the delay model is mismatched.



(a)

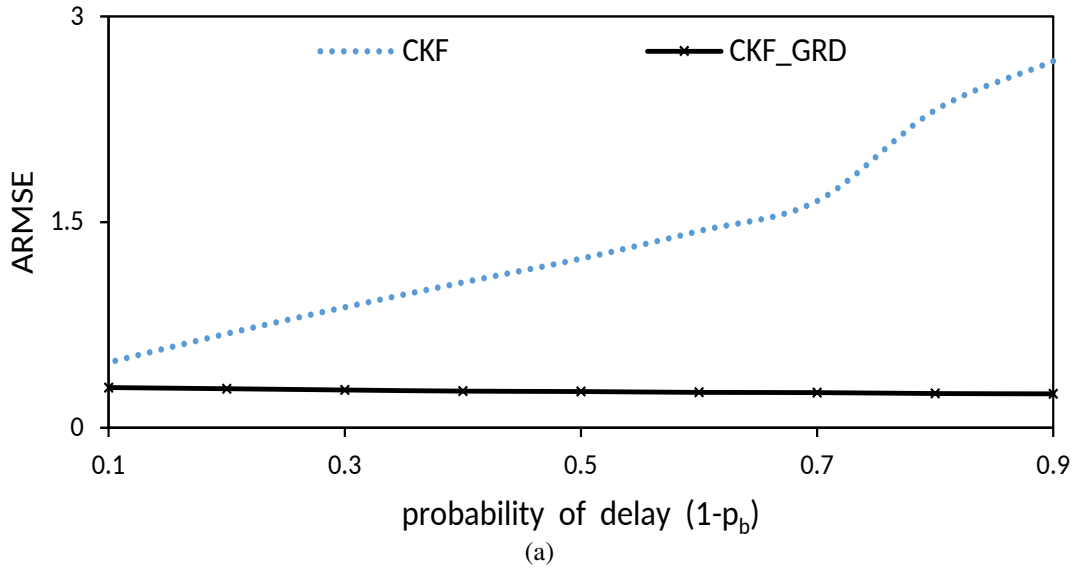


(b)

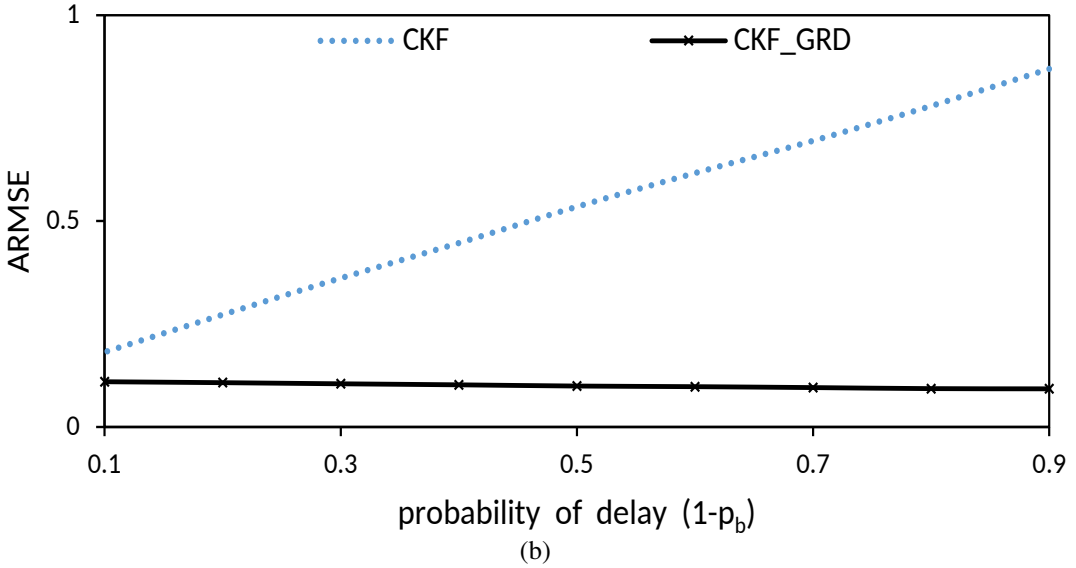
Figure 3.8: ARMSE analysis for mismatched delay model: (a) for Problem 1 with varying delay probability (p), and (b) for Problem 2 with varying delay probability (p).

3.4.4 Multi-sensor scenario

The performance analysis on different delays for different elements of measurement is provided in Fig. 3.9. The analysis is limited to the ARMSE. The ARMSE plot is drawn for Problem 2 by considering that only the second measurement is delayed. The ARMSE is reduced for the CKF_GRD compared to the ordinary CKF. Please note that MLCKF and CKF_RD are not applicable for varying delay across different elements; therefore, it is removed from the analysis in Fig. 3.9.



(a)



(b)

Figure 3.9: Problem 2: ARMSE plots of frequency and amplitude for varying delay probabilities across different elements of measurements for fixed $p_g = 0.3$: (a) frequency and (b) amplitude.

3.5 Summary

As the practical measurements are commonly delayed, an advanced Gaussian filtering method to encounter the delay is needed. The preliminary extensions of Gaussian filtering for delayed measurements suffer from several drawbacks, such as the inability to encounter large delay and arbitrary selection of the upper-bound of delay. Consecutively, they suffer from poor accuracy if the delay is large and the upper bound is unknown. Thus, this chapter introduces a new extension of the Gaussian filtering to handle delayed measurements. The proposed method reformulates the measurement model to introduce the delay possibility and redesigns the traditional Gaussian filtering for the reformulated measurement model. Unlike the existing delay filters, the proposed delay measurement model does not require apriori knowledge of the individual delay probabilities, which are mostly unknown or inaccurately known. Moreover, the proposed algorithm is generic and applicable to any of the existing Gaussian filters, such as the EKF, UKF, and CKF. The simulation results validate the improved accuracy of the proposed method. The primary limitation of this development is its assumption that delays are integer multiples of the sampling interval and that measurement noise follows a Gaussian distribution. However, in real-world scenarios, delays may often be a fraction of the sampling interval, and measurement noise tends to be non-Gaussian in nature.

Chapter 4

Fractionally Delayed Bayesian Approximation Filtering under Non-Gaussian Noisy Environment

4.1 Introduction

Recalling the discussions in Chapter 1, the Gaussian filtering is traditionally designed with various assumptions and by ignoring several practical challenges. Consequently, in practical applications, the Gaussian filtering suffers from various limitations [22]. Some of the popular limitations include the inefficacy of the Gaussian filtering in handling the non-Gaussian noises [19], delayed measurements [99], and intermittently missing measurements [100].

As discussed in Chapter 2, the literature witnesses various developments to handle the above-mentioned limitations individually. Yet, the literature lacks of an efficient solution for handling the joint occurrences of two or more of such limitations. This chapter introduces an advanced Gaussian filtering algorithm for jointly occurring delayed measurements and non-Gaussian measurement noises.

In Chapter 2, we reviewed various MC criterion-based Gaussian filters [157, 158, 123, 125] for handling the non-Gaussian noises (please see Section 2.3.3). We also reviewed various developments [92, 95, 97, 99, 159] in the Gaussian filtering for handling the delayed measurements, which are mostly designed under the MMSE criterion please see Section 2.3.1. In summary, their detailed

review infers the following: i) the existing MC criterion-based Gaussian filters [157, 158, 123, 125] underperform for the delayed measurements and ii) the MMSE criterion-based delayed Gaussian filters [92, 95, 97, 99, 159] underperform for the non-Gaussian noises. To the best of the author's knowledge, the existing Gaussian filters underperform in jointly handling the fractional delayed measurements and non-Gaussian measurement noises.

In this chapter, we design a new Gaussian filtering technique under the MC criterion to handle the non-Gaussian noises and delayed measurements jointly. We abbreviate the proposed method as GFMCFD, denoting Gaussian filtering under the MC criterion for fractionally delayed measurements. Interestingly, the GFMCFD handles fractional delays (delay being a fraction of the sampling interval), where the existing delay filters [92, 97, 99, 159] mostly underperform. The GFMCFD assigns multiple intermediate instants between every set of consecutive sampling instants in order to represent the fractional delays. Then, to handle the delay, the GFMCFD utilizes a Gaussian likelihood-based delay identification to analytically identify the delay in measurements. After the delay is identified, an appropriate past instant is explicitly known from where the current measurement arrives due to the delay. It is worth mentioning that this past instant may be a past intermediate time instant as well. Due to the delay, the current measurement carries appropriate state information corresponding to this explicitly identified past instant instead of the current instant. Therefore, the GFMCFD utilizes the current measurement to obtain the estimates of the state at this past instant instead of the current instant. Then, in order to perform the real-time state estimation, the GFMCFD updates the state estimates (corresponding to the past instant) till the current time instant using the process model. Finally, as mentioned previously, the complete design is performed under the MC criterion, which enables the GFMCFD to handle the non-Gaussian noises. As discussed above, to handle the delay, the current measurement is utilized to obtain the estimate of state at a stochastically identified past instant. Therefore, in our MC criterion-based design strategy, the cost function is defined at this particular past instant instead of the current instant. We validate the improved accuracy of the proposed method for two nonlinear filtering problems.

Summarizing the above discussions, we highlight the following novelties and contributions of the GFMCFD:

- In this study, the Gaussian filter jointly handles the delayed measurements and non-Gaussian measurement noises.

- It addresses fractional delay (a multiple of a fraction of the sampling interval), while the

existing delay filters either ignore the delays or simply address the integer delays.

- The filtering process is conducted by employing the intermediate of sampling instants, which is the preliminary specified instants between two sampling instants.
- It uses the current measurement to update the state at a past intermediate instant, while the existing filters generally use it for the current instant or, in some cases, for a past sampling instant (not an intermediate instant).
- Its cost function is different from the existing filters, as the cost function is defined at a past intermediate instant to handle the delay. Thus, the filter design requires a comprehensively different derivation.

4.2 Problem Formulation

Recalling Chapter 1, a nonlinear dynamical system can be represented by the following state space model,

$$\mathbf{x}_k = \phi_{k-1}(\mathbf{x}_{k-1}) + \nu_{k-1}, \tag{4.1}$$

$$\mathbf{z}_k = \Psi_k(\mathbf{x}_k) + \eta_k, \tag{4.2}$$

Please refer to Chapter 1 for more details.

To this end, let us consider the following notations: i) $\tilde{\mathbf{z}}_k$: a hypothetical non-delayed measurement with Gaussian noises, ii) $\mathbf{z}_k$: a hypothetical non-delayed measurement with non-Gaussian noises (same as Eq. (4.2)), iii) $\tilde{\mathbf{y}}_k$: a hypothetical delayed measurement with Gaussian noises, and iv) $\mathbf{y}_k$: the actually received measurement due to the fractional delay and non-Gaussian noises. Then, we infer the following from Chapter 1 and 2: i) the traditional Gaussian filters [48, 34] estimate $\mathbf{x}_k$ from $\tilde{\mathbf{z}}_k$, ii) the existing MC criterion-based Gaussian filters [123, 158] estimate $\mathbf{x}_k$ from $\mathbf{z}_k$, and iii) the existing delayed Gaussian filters [99, 159] estimate $\mathbf{x}_k$ from $\tilde{\mathbf{y}}_k$.

Summarizing the above discussion, the traditional Gaussian filtering method and its extensions fail or underperform in estimating $\mathbf{x}_k$ from the irregular measurement $\mathbf{y}_k$. We aim to design a new Gaussian filtering technique to estimate $\mathbf{x}_k$ from $\mathbf{y}_k$. We show the process of receiving the likely irregular measurement $\mathbf{y}_k$ in Fig. 4.1, considering that the delay appears due to the transmission lag [86, 90].

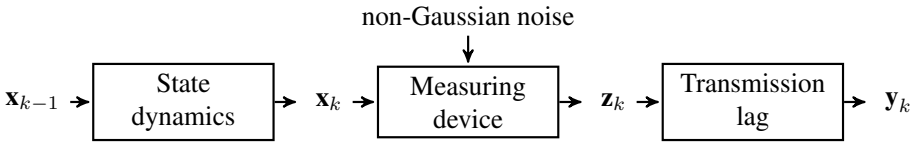


Figure 4.1: Block diagram of delay occurrence under non-Gaussian noisy environment. Here, the delay is appearing due to transmission lag with $\mathbf{z}_k$ and $\mathbf{y}_k$ as a hypothetical measurement without delay and the received measurement with delay, respectively.

From the above discussion, the objective of this chapter is to design an advanced Gaussian filtering algorithm for handling delayed measurements and non-Gaussian noises jointly. We take the following approach to accomplish this objective: i) we stochastically identify the delay, and restructure the traditional filtering methodology accordingly, to handle the delay and ii) we design the proposed filter under the MC criterion to address the non-Gaussian noises. Before proceeding to the next section, we briefly discussed the concept of correntropy.

4.2.1 Correntropy

According to information theory, correntropy is a measure of similarity between two probability distribution functions (PDFs)[119, 122]. The concept of entropy clearly relates to the concept of information content in PDFs. If there are finite numbers of samples available in which a correntropy between two random variables X and Y can be defined as

$$V(X,Y) = \mathbb{E}(G_\sigma(X-Y)) = \frac{1}{L}\sum_{j=1}^{L} G_\sigma(X_j - Y_j) \tag{4.3}$$

where L is the number of samples of the random variables, $G_\sigma(\cdot)$ is a kernel for the random variables in the sample valued; σ represents kernel bandwidth.

As a consequence of correntropy, the correntropy-induced metric (CIM) can be used to measure the similarity of two PDFs. There are a variety of ways to compute this metric, depending on the kernel and the bandwidth. In this way, it can be tuned to become insensitive to outliers, pushing it toward zero norms (indifference in distance).

The CIM distance ultimately motivates the use of the maximum correntropy criterion to make inferences [113]. In regression analysis, it searches for parameter values that maximize similarity among observations, *i.e.*,

$$\hat{\Theta} = \underset{\Theta\in\mho}{\arg\max}\frac{1}{L}\sum_{j=1}^{L} G_\sigma(e_j) \tag{4.4}$$

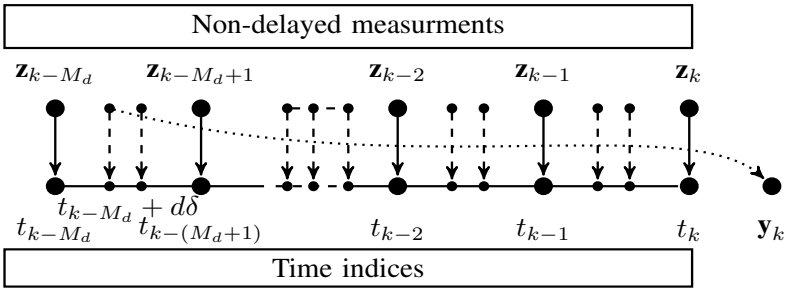


Figure 4.2: Schematic diagram of fractionally delayed measurement. Due to the delay, the receive measurement at t_k (denoted as $\mathbf{y}_k$) as $\mathbf{z}_{k-M_d+d\delta}$ instead of $\mathbf{z}_k$.

where $e_j = X_j - Y_j$, Θ is optimal parameter value, and $\mho$ is the feasible set of parameters.

4.3 Design Methodology of the GFMCFD

In this section, we introduce the design methodology of the GFMCFD for handling the fractionally delayed measurements under the non-Gaussian noisy environment. We assume that the maximum delay is restricted by M_d sampling intervals. We also follow the standard hypothesis that the process noises can be approximated as Gaussian, while the measurement noises are non-Gaussian [123, 125]. We characterize the fractional delays by hypothetically assigning M_s intermediate instants between two successive sampling instants (of the measurement data). Thus, if $\tau = t_k - t_{k-1}$ denotes the sampling interval, then the time-interval between two successive intermediate instants is $\delta = \tau/M_s$. We denote the d^{th} ($\forall d \in \{1,2,\cdots,M_s\}$) intermediate instant between t_{k-1} and t_k as $t_{k-1}^d = t_{k-1} + d\delta$. For $d = M_s$, we get $t_{k-1}^{M_s} = t_k$. Please refer to Fig. 4.2 for more clarity on the intermediate instants and the fractional delays.

In the remaining parts of this chapter, we simplify the notations $\mathbb{M}_{t_k+d\delta}$ and $\mathbb{M}_{t_k+d\delta|t_{k'}}$ as $\mathbb{M}_{k,d}$ and $\mathbb{M}_{k,d|k'}$, respectively, where $\mathbb{M}$ denotes a general symbolic parameter.

Being an extension of the Gaussian filtering, the GFMCFD involves intractable integrals and utilizes sample points and weights for approximating them numerically. Following the notations defined in Chapter 1, the sample points and associated weights are denoted as Υ_i and W_i, respectively $\forall i \in \{1,2,\cdots,N_s\}$, with N_s representing the number of sample points. In the subsequent parts, we discuss the design aspects of the prediction and update steps of the GFMCFD.

4.3.1 Prediction

As discussed in Chapter 1, the prediction step determines the prior estimate and covariance, $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$, respectively [48, 158] using the state dynamics under the Gaussian approximation of process noises. As the state dynamics is independent of the measurements, the prediction step remains unchanged for the delay occurrences (in the measurements) as well as for the non-Gaussian measurement noises. Henceforth, we will observe in the later discussions that the GFM-CFD requires the prior estimate and covariance at every past intermediate instant. Therefore, the GFMCFD determines the prior estimate and covariance through the intermediate instants, unlike the ordinary Gaussian filtering, which determines them only at the sampling instants.

The prior estimate and covariance at the intermediate instant t_{k-1}^{d}, *i.e.*, $\hat{\mathbf{x}}_{k-1,d|k-1}$ and $\mathbf{P}_{k-1,d|k-1}$, are obtained as

$$\begin{cases} \hat{\mathbf{x}}_{k-1,d|k-1} = \sum_{i=1}^{N_s} W_i \phi(\Upsilon_{i,(k-1,d-1)|k-1}), \\ \mathbf{P}_{k-1,d|k-1} = \sum_{i=1}^{N_s} W_i \Big(\phi(\Upsilon_{i,(k-1,d-1)|k-1}) - \hat{\mathbf{x}}_{k-1,d|k-1}\Big) \Big(\phi(\Upsilon_{i,(k-1,d-1)|k-1}) \\ \qquad\qquad -\hat{\mathbf{x}}_{k-1,d|k-1}\Big)^{T} + \mathbf{Q}_{k-1,d}, \end{cases} \tag{4.5}$$

where

$$\phi(\Upsilon_{i,(k-1,d-1)|k-1}) = \phi(\boldsymbol{\Sigma}_{k-1,d-1|k-1}\Upsilon_i + \hat{\mathbf{x}}_{k-1,d-1|k-1}),$$

with $\boldsymbol{\Sigma}_{k-1,d-1|k-1}$ representing the Cholesky decomposition of $\mathbf{P}_{k-1,d-1|k-1}$. Please note that

$$\begin{cases} \hat{\mathbf{x}}_{k-1,0|k-1} = \hat{\mathbf{x}}_{k-1|k-1}, \\ \hat{\mathbf{x}}_{k-1,M_s|k-1} = \hat{\mathbf{x}}_{k|k-1}, \\ \mathbf{P}_{k-1,0|k-1} = \mathbf{P}_{k-1|k-1}, \\ \mathbf{P}_{k-1,M_s|k-1} = \mathbf{P}_{k|k-1}. \end{cases} \tag{4.6}$$

To this end, for maximum delay M_d, let us define the following: i) $t_g \in \{t_{k-M_d}, t_{k-M_d+1}, \cdots, t_{k-1}\}$: a particular past sampling instant from the set of all past sampling instants within the range of maximum delay and ii) $N_t \in \{t_{k-M_d}+d\delta, t_{k-M_d+1}+d\delta \cdots, t_{k-1}+d\delta\}\ \forall d \in \{1,2,\cdots,M_s\}$: a particular intermediate instant from all past intermediate instances within the range of maximum delay. Then, we can denote $\hat{\mathbf{x}}_{g,d|k-1}$ and $\mathbf{P}_{g,d|k-1}$ $\forall g \in \{k-M_d, k-M_d+1, \cdots, k-1\}$ and

$\forall d \in \{1,2,\cdots,M_s\}$ as the sets of all prior estimate and covariance (Eq. (4.5)), respectively till the current time instant.

4.3.2 Update

The update step determines the posterior estimate and covariance, $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$, respectively, as the irregular measurement $\mathbf{y}_k$ is received at t_k. As discussed previously, $\mathbf{y}_k$ consists of fractional delays and non-Gaussian noises. Due to the delay, $\mathbf{y}_k$ carries the state information related to a past instant t_g^d instead of the current instant t_k. Thus, $\mathbf{y}_k$ should be more appropriate to estimate the state $\mathbf{x}$ at the past (delayed) instant t_g^d instead of t_k. Please note that the delayed instance t_g^d is unknown. We use the maximum likelihood approach to identify the delay and determine the appropriate delayed instance t_g^d for $\mathbf{y}_k$. Following this, we utilize $\mathbf{y}_k$ to estimate the states at the past instance t_g^d. We finally update the estimated state from t_g^d to t_k (the current instant) using the state dynamics to perform a real-time state estimation.

Summarizing the above discussion, the update step involves three sub-steps: (i) delay identification, (ii) determining posterior estimates at the past instant t_g^d, and (iii) updating the posterior estimates from t_g^d to t_k. In the subsequent discussions, we describe the computational aspects of the three sub-steps.

Delay identification

We utilize negative Gaussian log-likelihood to identify the delay [3]. In this regard, let us denote $\hat{\mathbf{y}}_{g,d|k-1}$ and $\mathbf{P}^{\mathbf{yy}}_{g,d|k-1}$ as the predicted estimate and covariance of the measurement, respectively, at the past instant t_g^d. Extending the general Gaussian filtering methodology of determining $\hat{\mathbf{y}}_{k|k-1}$ and $\mathbf{P}^{\mathbf{yy}}_{k|k-1}$ at t_k, we obtain $\hat{\mathbf{y}}_{g,d|k-1}$ and $\mathbf{P}^{\mathbf{yy}}_{g,d|k-1}$ at t_g^d as

$$\hat{\mathbf{y}}_{g,d|k-1} = \sum_{i=1}^{N_s} W_i \Psi(\Upsilon_{i,(g,d)|k-1}), \tag{4.7}$$

$$\mathbf{P}^{\mathbf{yy}}_{g,d|k-1} = \sum_{i=1}^{N_s} W_i \Big(\Psi(\Upsilon_{i,(g,d)|k-1}) - \hat{\mathbf{y}}_{g,d|k-1}\Big)\Big(\Psi(\Upsilon_{i,(g,d)|k-1}) - \hat{\mathbf{y}}_{g,d|k-1}\Big)^T + \mathbf{R}_{g,d}, \tag{4.8}$$

where $\Psi(\Upsilon_{i,(g,d)|k-1}) = \Psi(\mathbf{\Sigma}_{g,d|k-1}\Upsilon_i + \hat{\mathbf{x}}_{g,d|k-1})$.

Although the Gaussian likelihood of $\mathbf{y}_k$ arriving from the past instant t_g^d, denoted as $L_{g,d}(\mathbf{y}_k)$,

can be given as

$$L_{g,d}(\mathbf{y}_k) \sim \frac{1}{(2\pi)^r det(\mathbf{P}^{\mathbf{yy}}_{g,d|k-1})} \exp\Big(-\mathbb{B}_{g,d|k-1}\Big), \tag{4.9}$$

where $\mathbb{B}_{g,d|k-1} = (\mathbf{y}_k - \hat{\mathbf{y}}_{g,d|k-1})(\mathbf{P}^{\mathbf{yy}}_{g,d|k-1})^{-1}(\mathbf{y}_k - \hat{\mathbf{y}}_{g,d|k-1})^T$.

Further, the Gaussian log-likelihood of $\mathbf{y}_k$ arriving from t_g^d, denoted as $\mathbb{L}_{g,d}(\mathbf{y}_k)$, can be expressed as

$$\mathbb{L}_{g,d}(\mathbf{y}_k) \sim -\log\Big(det(\mathbf{P}^{\mathbf{yy}}_{g,d|k-1})\Big) - \mathbb{B}_{g,d|k-1}. \tag{4.10}$$

In the above likelihood expression, we ignored the constant terms, as the likelihoods are meant for numeric comparison, which is uninfluenced by constant terms. For similar reasons, and as a further simplification, we take the negative Gaussian log-likelihood of $\mathbf{y}_k$ arriving from t_g^d as

$$\mathbb{L}_{g,d}(\mathbf{y}_k) \sim \log\Big(det(\mathbf{P}^{\mathbf{yy}}_{g,d|k-1})\Big) + \mathbb{B}_{g,d|k-1}. \tag{4.11}$$

Remark 4.3.1 *As $\mathbb{L}_{g,d}(\mathbf{y}_k)$ gives negative likelihood, the minimum of $\mathbb{L}_{g,d}(\mathbf{y}_k)$ gives the maximum likelihood of $\mathbf{y}_k$ arriving from t_g^d, and vice versa.*

We obtain the negative likelihood $\mathbb{L}_{g,d}(\mathbf{y}_k)$ $\forall g$ and d, *i.e.*, for all past intermediate instants till t_{k-M_d}. Thereafter, we obtain a past instant $t_{\tilde{g}}^{\tilde{d}} = t_{\tilde{g}} + \tilde{d}\delta$ from where $\mathbf{y}_k$ is most likely to arrive due to the delay by determining $(\tilde{g}, \tilde{d})$, as

$$(\tilde{g}, \tilde{d}) = \arg\min_{g,d} \mathbb{L}_{g,d}(\mathbf{y}_k). \tag{4.12}$$

As we obtain $(\tilde{g}, \tilde{d})$, we can state that $\mathbf{y}_k$ would have been received at the past instant $t_{\tilde{g}}^{\tilde{d}}$, if there was no delay. Thus, $\mathbf{y}_k$ should be most appropriate for estimating $\mathbf{x}$ at $t_{\tilde{g}}^{\tilde{d}}$ instead of t_k. Consequently, we utilize $\mathbf{y}_k$ to estimate $\mathbf{x}$ at $t_{\tilde{g}}^{\tilde{d}}$.

State estimation at the past instant $t_{\tilde{g}}^{\tilde{d}}$

In this step, we determine $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k}$ and $\mathbf{P}_{\tilde{g},\tilde{d}|k}$ at the past instant $t_{\tilde{g}}^{\tilde{d}}$ from which $\mathbf{y}_k$ is most likely to arrive at t_k due to the delay. The computation of $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k}$ and $\mathbf{P}_{\tilde{g},\tilde{d}|k}$ requires $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k-1}$, $\mathbf{P}_{\tilde{g},\tilde{d}|k-1}$, $\hat{\mathbf{y}}_{\tilde{g},\tilde{d}|k-1}$, $\mathbf{P}^{\mathbf{yy}}_{\tilde{g},\tilde{d}|k-1}$, $\mathbf{P}^{\mathbf{xy}}_{\tilde{g},\tilde{d}|k-1}$. We obtain $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k-1}$, $\mathbf{P}_{\tilde{g},\tilde{d}|k-1}$, $\hat{\mathbf{y}}_{\tilde{g},\tilde{d}|k-1}$, and $\mathbf{P}^{\mathbf{yy}}_{\tilde{g},\tilde{d}|k-1}$ through Eqs. (4.5), (4.7), and (4.8). Moreover, we determine $\mathbf{P}^{\mathbf{xy}}_{\tilde{g},\tilde{d}|k-1}$ as

$$\mathbf{P}^{\mathbf{xy}}_{\tilde{g},\tilde{d}|k-1} = \sum_i W_i \Big(\Upsilon_{i,(\tilde{g},\tilde{d})|k-1} - \hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k-1} \Big) \Big(\Psi(\Upsilon_{i,(\tilde{g},\tilde{d})|k-1}) - \hat{\mathbf{y}}_{\tilde{g},\tilde{d}|k-1} \Big)^T. \tag{4.13}$$

As discussed previously, we use the MC criterion for the filter design in order to handle the non-Gaussian measurement noises. Thus, the further steps of determining $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k}$ and $\mathbf{P}_{\tilde{g},\tilde{d}|k}$ follows the MC criterion unlike the MMSE criterion used in the traditional Gaussian filtering. In this design criterion, we consider an augmented error for augmented state and measurement, unlike the MMSE criterion which considers the errors only for the states. Thus, the desired estimate $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k}$ is obtained by solving the regression model of augmented states [123, 158], *i.e.,*

$$\begin{bmatrix} \hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k-1} \\ \mathbf{y}_k \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{\tilde{g},\tilde{d}} \\ \Psi_k(\mathbf{x}_{\tilde{g},\tilde{d}}) \end{bmatrix} + \boldsymbol{\varpi}_{\tilde{g},\tilde{d}}, \tag{4.14}$$

where $\boldsymbol{\varpi}_{\tilde{g},\tilde{d}} \in \mathbb{R}^{n+r}$ is the augmented error, given as $\boldsymbol{\varpi}_{\tilde{g},\tilde{d}} = [-\boldsymbol{\xi}^T_{\tilde{g},\tilde{d}} \quad \boldsymbol{\eta}^T_{\tilde{g},\tilde{d}}]^T$, with $\boldsymbol{\xi}_{\tilde{g},\tilde{d}} = \mathbf{x}_{\tilde{g},\tilde{d}} - \hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k-1}$.

To this end, our objective is to solve the regression model (4.14) in order to obtain the estimate $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}}$ at the past instant $t_{\tilde{g}}^{\tilde{d}}$. The regression model, however, involves a nonlinear function $\Psi_k(\mathbf{x}_{\tilde{g},\tilde{d}})$, making the solutions of regression model non-trivial. Therefore, before proceeding further, we linearize $\Psi_k(\mathbf{x}_{\tilde{g},\tilde{d}})$ in order to ultimately linearize the regression model. The linearization of the measurement model can be achieved through derivative-based linearization or statistical linearization. The derivative-based linearization, however, causes large residual errors and poor numerical stability. Therefore, we choose statistical linearization for better numerical stability and smaller residual error. Therefore, before proceeding further, we linearize the measurement model using the statistical linearization method [60, 158], giving

$$\mathbf{y}_k = \hat{\mathbf{y}}_{\tilde{g},\tilde{d}|k-1} + \mathbb{H}_{\tilde{g},\tilde{d}} \boldsymbol{\xi}_{\tilde{g},\tilde{d}} + \boldsymbol{\eta}_{\tilde{g},\tilde{d}} + \vartheta_{\tilde{g},\tilde{d}}, \tag{4.15}$$

where $\vartheta_{\tilde{g},\tilde{d}}$ represents the linearization error and $\mathbb{H}_{\tilde{g},\tilde{d}} = [\mathbf{P}^{-1}_{\tilde{g},\tilde{d}|k-1}\mathbf{P}^{\mathbf{xy}}_{\tilde{g},\tilde{d}|k-1}]^T$ denotes the linearization matrix. The reader may refer to [60, 158] for a detailed discussion on the statistical linearization.

After substituting $\mathbf{y}_k$ from Eq. (4.15) into Eq. (4.14), we get

$$\begin{bmatrix} \hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k-1} \\ \mathbf{y}_k - \hat{\mathbf{y}}_{\tilde{g},\tilde{d}|k-1} + \mathbb{H}_{\tilde{g},\tilde{d}}\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k-1} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_n \\ \mathbb{H}_{\tilde{g},\tilde{d}} \end{bmatrix} \mathbf{x}_{\tilde{g},\tilde{d}} + \psi_{\tilde{g},\tilde{d}}, \tag{4.16}$$

where $\mathbf{I}_n$ denotes the n-dimensional identity matrix and $\psi_{\tilde{g},\tilde{d}} = [-\xi^T_{\tilde{g},\tilde{d}} \quad \eta^T_{\tilde{g},\tilde{d}} + \vartheta^T_{\tilde{g},\tilde{d}}]^T$ is the augmented error. Then, the augmented error covariance can be given as [123, 158]

$$E(\psi_{\tilde{g},\tilde{d}}\psi^T_{\tilde{g},\tilde{d}}) = \Omega_{\tilde{g},\tilde{d}}\Omega^T_{\tilde{g},\tilde{d}} = \begin{bmatrix} \Omega^n_{\tilde{g},\tilde{d}}\Omega^{nT}_{\tilde{g},\tilde{d}} & \mathbf{0} \\ \mathbf{0} & \Omega^r_{\tilde{g},\tilde{d}}\Omega^{rT}_{\tilde{g},\tilde{d}} \end{bmatrix}, \tag{4.17}$$

and

$$\begin{cases} \Omega_{\tilde{g},\tilde{d}} = chol\big(\mathbb{E}(\psi_{\tilde{g},\tilde{d}}\psi^T_{\tilde{g},\tilde{d}})\big), \\ \Omega^n_{\tilde{g},\tilde{d}} = chol\big(\mathbf{P}_{\tilde{g},\tilde{d}|k-1}\big), \\ \Omega^r_{\tilde{g},\tilde{d}} = chol\big(\mathbb{E}((\eta_{\tilde{g},\tilde{d}} + \vartheta_{\tilde{g},\tilde{d}})(\eta_{\tilde{g},\tilde{d}} + \vartheta_{\tilde{g},\tilde{d}})^T)\big), \\ \qquad\quad = chol\big(\mathbf{P}^{\mathbf{yy}}_{\tilde{g},\tilde{d}|k-1} - \mathbf{P}^{\mathbf{xy}T}_{\tilde{g},\tilde{d}|k-1}\mathbf{P}^{-1}_{\tilde{g},\tilde{d}|k-1}\mathbf{P}^{\mathbf{xy}}_{\tilde{g},\tilde{d}|k-1}\big). \end{cases} \tag{4.18}$$

We normalize the regression model given in Eq. (4.16) by multiplying $\Omega^{-1}_{\tilde{g},\tilde{d}}$ in both sides, which gives

$$\mathbf{D}_{\tilde{g},\tilde{d}} = \mathbf{W}_{\tilde{g},\tilde{d}}\mathbf{x}_{\tilde{g},\tilde{d}} + \mathbf{e}_{\tilde{g},\tilde{d}}, \tag{4.19}$$

where

$$\begin{cases} \mathbf{D}_{\tilde{g},\tilde{d}} = \Omega^{-1}_{\tilde{g},\tilde{d}} \begin{bmatrix} \hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k-1} \\ \mathbf{y}_k - \hat{\mathbf{y}}_{\tilde{g},\tilde{d}|k-1} + \mathbb{H}_{\tilde{g},\tilde{d}}\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k-1} \end{bmatrix}, \\ \mathbf{W}_{\tilde{g},\tilde{d}} = \Omega^{-1}_{\tilde{g},\tilde{d}} \begin{bmatrix} \mathbf{I}_n \\ \mathbb{H}_{\tilde{g},\tilde{d}} \end{bmatrix}, \\ \mathbf{e}_{\tilde{g},\tilde{d}} = \Omega^{-1}_{\tilde{g},\tilde{d}} \begin{bmatrix} -\xi_{\tilde{g},\tilde{d}} \\ \eta_{\tilde{g},\tilde{d}} + \vartheta_{\tilde{g},\tilde{d}} \end{bmatrix}. \end{cases} \tag{4.20}$$

Please note that $\mathbf{D}_{\tilde{g},\tilde{d}}$, $\mathbf{W}_{\tilde{g},\tilde{d}}$, and $\mathbf{e}_{\tilde{g},\tilde{d}}$ are $(n+r)$-dimensional matrices. In the later parts, we denote

$D_{i,(\tilde{g},\tilde{d})}$, $W_{i,(\tilde{g},\tilde{d})}$, and $e_{i,(\tilde{g},\tilde{d})}$ as i^{th} elements of $\mathbf{D}_{\tilde{g},\tilde{d}}$, $\mathbf{W}_{\tilde{g},\tilde{d}}$, and $\mathbf{e}_{\tilde{g},\tilde{d}}$, respectively $\forall i \in \{1,2,\cdots,n+r\}$.

Under the MC criterion-based filter design, an optimal estimate of state maximizes the cost function [119, 158]

$$\mathscr{J}_{n+r} = \sum_{j=1}^{n+r} G_\sigma(e_{j,(\tilde{g},\tilde{d})}) - \frac{\kappa}{2} e^2_{j,(\tilde{g},\tilde{d})}, \tag{4.21}$$

where $G_\sigma(e_{j,(\tilde{g},\tilde{d})}) = exp(-e^2_{j,(\tilde{g},\tilde{d})}/2\sigma^2)$ denotes Gaussian kernel with bandwidth σ, while κ represents the regularization parameter.

Remark 4.3.2 *Eq.* (4.21) *defines the cost function at the past intermediate instant $t_{\tilde{g}}^{\tilde{d}}$, unlike the exiting filters, defining it at the current instant t_k.*

Remark 4.3.3 *The maximum correntropy cost function Eq.* (4.4) *is ill-posed [158]. To tackle this issue a regularization term is added to the Eq.* (4.21).

For the given cost function $\mathscr{J}_{n+r}$, we obtain the optimal estimate at $t_{\tilde{g}}^{\tilde{d}}$, *i.e.*, $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k}$, as the solution of $\frac{\partial \mathscr{J}_{n+r}}{\partial \mathbf{x}_{\tilde{g},\tilde{d}}} = 0$, giving

$$\sum_{j=1}^{n+r} e_{j,(\tilde{g},\tilde{d})} G_\sigma(e_{j,(\tilde{g},\tilde{d})}) W^T_{j,(\tilde{g},\tilde{d})} + \kappa e_{j,(\tilde{g},\tilde{d})} W^T_{j,(\tilde{g},\tilde{d})} = 0. \tag{4.22}$$

After further simplifications, we get [119, 158]

$$\mathbf{x}_{\tilde{g},\tilde{d}} = \left[\sum_{j=1}^{n+r} \big(G_\sigma(e_{j,(\tilde{g},\tilde{d})}) + \hat{\kappa}\big) W_{j,(\tilde{g},\tilde{d})} W^T_{j,(\tilde{g},\tilde{d})}\right]^{-1} \left[\sum_{j=1}^{n+r} \big(G_\sigma(e_{j,(\tilde{g},\tilde{d})}) + \hat{\kappa}\big) W^T_{j,(\tilde{g},\tilde{d})} d_{j,(\tilde{g},\tilde{d})}\right], \tag{4.23}$$

where $\hat{\kappa} = \sigma^2\kappa/\tilde{\beta}$ and $\tilde{\beta} = 1/\sqrt{2\pi\sigma}$. Please note that an analytical solution of Eq. (4.23) is non-trivial. Thus, we use fixed-point iterative method to determine its solution and obtain the desired estimate $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k}$.

Before discussing the iterative method, let us define $\mathbb{S}_{\tilde{g},\tilde{d}} = diag\Big(\mathbb{S}_{x,(\tilde{g},\tilde{d})}, \mathbb{S}_{z,(\tilde{g},\tilde{d})}\Big)$, where

$$\begin{cases} \mathbb{S}_{x,(\tilde{g},\tilde{d})} = diag\Big(G_\sigma\big(e_{1,(\tilde{g},\tilde{d})}\big) + \hat{\kappa}, \cdots, G_\sigma\big(e_{n,(\tilde{g},\tilde{d})}\big) + \hat{\kappa}\Big), \\ \mathbb{S}_{z,(\tilde{g},\tilde{d})} = diag\Big(G_\sigma\big(e_{n+1,(\tilde{g},\tilde{d})}\big) + \hat{\kappa}, \cdots, G_\sigma\big(e_{n+r,(\tilde{g},\tilde{d})}\big) + \hat{\kappa}\Big), \end{cases} \tag{4.24}$$

with $diag(\cdot)$ representing a diagonal matrix for the given diagonal elements. Subsequently, we

rewrite Eq. (4.23) in iterative form, as

$$\hat{\mathbf{x}}^{t}_{\tilde{g},\tilde{d}} = \left(\mathbf{W}^{T}_{\tilde{g},\tilde{d}}\mathbb{S}^{t-1}_{\tilde{g},\tilde{d}}\mathbf{W}_{\tilde{g},\tilde{d}}\right)^{-1}\left(\mathbf{W}^{T}_{\tilde{g},\tilde{d}}\mathbb{S}^{t-1}_{\tilde{g},\tilde{d}}\mathbf{D}_{\tilde{g},\tilde{d}}\right), \tag{4.25}$$

where $\hat{\mathbf{x}}^{t}_{\tilde{g},\tilde{d}}$ denotes the optimal estimate of $\mathbf{x}_{\tilde{g},\tilde{d}}$ in t^{th} fixed-point iteration at $t^{\tilde{d}}_{\tilde{g}}$.

We further simplify Eq. (4.25) in Appendix A to obtain the iterative solution $\hat{\mathbf{x}}^{t}_{\tilde{g},\tilde{d}}$ in the form of a general estimate expression, as

$$\hat{\mathbf{x}}^{t}_{\tilde{g},\tilde{d}} = \hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k-1} + \mathbf{K}^{t-1}_{\tilde{g},\tilde{d}|k}(\mathbf{y}_k - \hat{\mathbf{y}}_{\tilde{g},\tilde{d}|k-1}), \tag{4.26}$$

where

$$\mathbf{K}^{t-1}_{\tilde{g},\tilde{d}|k} = \bar{\mathbf{P}}_{\tilde{g},\tilde{d}|k-1}\mathbb{H}^{T}_{\tilde{g},\tilde{d}}\left(\mathbb{H}_{\tilde{g},\tilde{d}}\bar{\mathbf{P}}_{\tilde{g},\tilde{d}|k-1}\mathbb{H}^{T}_{\tilde{g},\tilde{d}} + \bar{\mathbf{R}}^{t-1}_{\tilde{g},\tilde{d}|k}\right), \tag{4.27}$$

$$\bar{\mathbf{P}}_{\tilde{g},\tilde{d}|k-1} = \Omega^{n}_{\tilde{g},\tilde{d}}\mathbb{S}^{-1}_{x,(\tilde{g},\tilde{d})}\Omega^{nT}_{\tilde{g},\tilde{d}}, \tag{4.28}$$

$$\bar{\mathbf{R}}^{t-1}_{\tilde{g},\tilde{d}|k} = \Omega^{r}_{\tilde{g},\tilde{d}}\mathbb{S}^{-1}_{z,(\tilde{g},\tilde{d})}\Omega^{rT}_{\tilde{g},\tilde{d}}. \tag{4.29}$$

Similarly, extending the derivation used in [119, 158] to obtain the posterior covariance $\mathbf{P}_{k|k}$ at t_k, we get $\mathbf{P}^{t}_{\tilde{g},\tilde{d}}$ at $t^{\tilde{d}}_{\tilde{g}}$ as

$$\mathbf{P}^{t}_{\tilde{g},\tilde{d}} = \left(\mathbf{I}_n - \mathbf{K}^{t-1}_{\tilde{g},\tilde{d}|k}\mathbb{H}_{\tilde{g},\tilde{d}}\right)\mathbf{P}_{\tilde{g},\tilde{d}|k-1}\left(\mathbf{I}_n - \mathbf{K}^{t-1}_{\tilde{g},\tilde{d}|k}\mathbb{H}_{\tilde{g},\tilde{d}}\right)^{T} + \mathbf{K}^{t-1}_{\tilde{g},\tilde{d}|k}\mathbf{R}_{\tilde{g},\tilde{d}}(\mathbf{K}^{t-1}_{\tilde{g},\tilde{d}|k})^{T}. \tag{4.30}$$

To stop the iterative process, we define an error quantity,

$$\varepsilon = \frac{||\hat{\mathbf{x}}^{t}_{\tilde{g},\tilde{d}} - \hat{\mathbf{x}}^{t-1}_{\tilde{g},\tilde{d}}||}{||\hat{\mathbf{x}}^{t-1}_{\tilde{g},\tilde{d}}||}, \tag{4.31}$$

where $||\cdot||$ denotes the second norm. The iterative process is stopped as ε becomes smaller than a predefined tolerance level tol. At the end of the iterative process, we obtain the desired estimate and covariance, *i.e.*, $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k}$ and $\mathbf{P}_{\tilde{g},\tilde{d}|k}$, respectively, as

$$\begin{cases} \hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k} = \hat{\mathbf{x}}^{t}_{\tilde{g},\tilde{d}}, \\ \mathbf{P}_{\tilde{g},\tilde{d}|k} = \mathbf{P}^{t}_{\tilde{g},\tilde{d}}. \end{cases} \tag{4.32}$$

The pseudo code for determining $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k}$ and $\mathbf{P}_{\tilde{g},\tilde{d}|k}$ from $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k-1}$ and $\mathbf{P}_{\tilde{g},\tilde{d}|k-1}$ is provided in Algo-

rithm 1.

Algorithm 1 Estimating $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k}$ and $\mathbf{P}_{\tilde{g},\tilde{d}|k}$ from $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k-1}$ and $\mathbf{P}_{\tilde{g},\tilde{d}|k-1}$.

Input: $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k-1}$, $\mathbf{P}_{\tilde{g},\tilde{d}|k-1}$, tol, κ, and σ
Output: $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k}$ and $\mathbf{P}_{\tilde{g},\tilde{d}|k}$

Initialization: $\hat{\mathbf{x}}^0_{\tilde{g},\tilde{d}} = \hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k-1}$ and $t = 1$
Obtain $\Omega^n_{\tilde{g},\tilde{d}}$, $\Omega^r_{\tilde{g},\tilde{d}}$, and $\Omega_{\tilde{g},\tilde{d}}$ from Eq. (4.18)
while $\varepsilon > tol$ **do**
  Obtain $\mathbf{e}_{\tilde{g},\tilde{d}} = \mathbf{D}_{\tilde{g},\tilde{d}} - \mathbf{W}_{\tilde{g},\tilde{d}}\hat{\mathbf{x}}^{t-1}_{\tilde{g},\tilde{d}}$ using Eq. (4.19)
  Determine $\mathbb{S}_{x,(\tilde{g},\tilde{d})}$ and $\mathbb{S}_{z,(\tilde{g},\tilde{d})}$ using Eq. (4.24)
  Determine $\bar{\mathbf{P}}_{\tilde{g},\tilde{d}|k-1}$ and $\bar{\mathbf{R}}^{t-1}_{\tilde{g},\tilde{d}|k}$ using Eqs. (4.28) and 4.29, respectively
  Determine $\mathbf{K}^{t-1}_{\tilde{g},\tilde{d}|k}$ using Eq. (4.27)
  Determine the estimated value $\hat{\mathbf{x}}^t_{\tilde{g},\tilde{d}}$ using Eq. (4.26)
  Obtain the error quantity ε using Eq. (4.31)
  $t = t + 1$
end while
Determine $\mathbf{P}^t_{\tilde{g},\tilde{d}}$ using Eq. (4.30)
return $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k} = \hat{\mathbf{x}}^t_{\tilde{g},\tilde{d}}$ and $\mathbf{P}_{\tilde{g},\tilde{d}|k} = \mathbf{P}^t_{\tilde{g},\tilde{d}}$

Updating $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k}$ and $\mathbf{P}_{\tilde{g},\tilde{d}|k}$ from $t^{\tilde{d}}_{\tilde{g}}$ to t_k

In the previous step, we determined $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k}$ and $\mathbf{P}_{\tilde{g},\tilde{d}|k}$ at the past instant $t^{\tilde{d}}_{\tilde{g}}$. We now propagate $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k}$ and $\mathbf{P}_{\tilde{g},\tilde{d}|k}$ through the state dynamics from $t^{\tilde{d}}_{\tilde{g}}$ to t_k to obtain $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ for performing the real-time state estimation. In this regard, let us denote M_{int} as the number of intermediate instants between $t^{\tilde{d}}_{\tilde{g}}$ and t_k. Then, we can perform the following time-shifts for j varying from 1 to M_{int}.

$$\hat{\mathbf{x}}_{\tilde{g},\tilde{d}+j|k} = \sum_{i=1}^{N_s} W_i \phi\left(\Upsilon_{i,(\tilde{g},\tilde{d}+j-1)|k}\right), \tag{4.33}$$

$$\mathbf{P}_{\tilde{g},\tilde{d}+j|k} = \sum_{i=1}^{N_s} W_i \left(\phi(\Upsilon_{i,(\tilde{g},\tilde{d}+j-1)|k}) - \hat{\mathbf{x}}_{\tilde{g},\tilde{d}+j|k}\right)\left(\phi(\Upsilon_{i,(\tilde{g},\tilde{d}+j-1)|k}) - \hat{\mathbf{x}}_{\tilde{g},\tilde{d}+j|k}\right)^T + \mathbf{Q}_{\tilde{g},\tilde{d}+j}. \tag{4.34}$$

Please note that, for $j = 1$, $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}+j-1|k} = \hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k}$ and $\mathbf{P}_{\tilde{g},\tilde{d}+j-1|k} = \mathbf{P}_{\tilde{g},\tilde{d}|k}$, which are already known from the previous step. Moreover, for $j = M_{int}$, we obtain $\hat{\mathbf{x}}_{\tilde{g},M_{int}|k} = \hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{\tilde{g},M_{int}|k} = \mathbf{P}_{k|k}$, giving the desired estimate and covariance. The pseudo code for determining $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ from $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k}$ and $\mathbf{P}_{\tilde{g},\tilde{d}|k}$ is provided in Algorithm 2.

By using the filtering strategy outlined above, we can handle delayed measurements and non-Gaussian noise. This filtering approach includes fractional delay and an MC-based criterion within

Algorithm 2 Estimating $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ from $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k}$ and $\mathbf{P}_{\tilde{g},\tilde{d}|k}$.

Input: $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k}$, $\mathbf{P}_{\tilde{g},\tilde{d}|k}$
Output: $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$
Initialization: $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}+0|k} = \hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k}$, $\mathbf{P}_{\tilde{g},\tilde{d}+0|k} = \mathbf{P}_{\tilde{g},\tilde{d}|k}$, and $j = 1$
1: **for** $j = 1 : M_{int}$ **do**
2: Determine $\hat{\mathbf{x}}_{\tilde{g},\tilde{d}+j|k}$ using Eq. (4.33)
3: Determine $\mathbf{P}_{\tilde{g},\tilde{d}+j|k}$ using Eq. (4.34)
4: **end for**
5: **return** $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{\tilde{g},M_{int}|k}$ and $\mathbf{P}_{k|k} = \mathbf{P}_{\tilde{g},M_{int}|k}$

a Gaussian filtering framework, which sets it apart from existing delay filters that only operate under integer delay and MMSE-based criterion. One significant advantage of this approach is that it does not require stochastic models to estimate delays, which is not the case for the existing delay filters. However, the GFMCFD requires additional computational and storage budgets in comparison to the traditional Gaussian filters, as described in Algorithm I and II.

Similar to other MC criterion-based filters, the GFMCFD linearizes the measurement model (please see Eq. (4.15)). The early MC criterion-based filters used derivative-based linearization, causing high sensitivity to outliers, large residual error, and poor numerical stability. To mitigate these drawbacks, the GFMCFD uses the recently introduced statistical linearization [158].

Remark 4.3.4 *Selection of M_d and M_s are a practitioner's choice with the freedom to choose any large value with the realization of increased computational demand. Therefore, despite of M_d being a design parameter, the GFMCFD is not restricted in terms of maximum delay.*

Remark 4.3.5 *The GFMCFD performs filtering through intermediate instants and involves multiple sub-steps in the update step. Thus, its computational demand is relatively higher than the traditional Gaussian filtering.*

The GFMCFD is a general extension to the Gaussian filtering, which applies to any of the existing Gaussian filters, such as the UKF, CKF, and GHF. As the joint occurrences of the non-Gaussian measurement noises and delayed measurements are practically common, the GFMCFD diversifies the practical applicability of the Gaussian filtering. Hence, we can further enhance the GFMCFD to address more irregularities, such as the missing measurements and cyber-attacks.

4.4 Simulation and Results

In this section, we simulate the CKF-based formulation of the GFMCFD, abbreviated as MC-CKF_FD, for two nonlinear filtering problems. Subsequently, we compare the performance of the MCCKF_FD with the traditional CKF and the CKF-based formulations of various benchmark filtering methods, particularly [123], [92], [97], [99], and [159]. The CKF-based formulations of [123], [92], [97], [99], and [159] are abbreviated as MCCKF, CKF_1RD, CKF_RD, MLCKF, and CKF_GRD, respectively. These benchmark filtering methods handle the delays of different natures. The simulation is performed in Matlab 2020 in a personal computer with configuration Intel i5, 3.40GHz processor, and 32GB RAM, as mentioned in the Chapter 1. The performance analysis is based on average root mean square error (ARMSE). As discussed in Chapter 1, we choose the RMSE as the performance analysis. More appropriately, we choose ARMSE, as defined in the previous chapter.

4.4.1 Simulating irregular measurement data $\mathbf{z}_k$

In the lack of real-data, we use Matlab simulated data-based analysis, which is commonly accepted in the literature [97, 99]. To simulate the irregular measurement data $\mathbf{y}_k \ \forall k \in \{1,2,\cdots\}$, we model $\mathbf{y}_k$ in terms of hypothetical non-delayed measurement $\mathbf{z}_{g,d}$ (Eq. (4.2)), as

$$\mathbf{y}_k = \sum_{g=k-M_d}^{k-1} \sum_{d=1}^{M_s} \alpha_{g,d} \mathbf{z}_{g,d}, \tag{4.35}$$

where $\alpha_{g,d}$, with $g \in \{k-M_d, k-M_d+1, \cdots, k-1\}$ and $d \in \{1,2,\cdots,M_s\}$, denotes an array of Bernoulli random variables. At any time t_k, only one entry of $\alpha_{g,d}$ can be unity, while the others are necessarily zero. In continuation, $\mathbf{y}_k$ simply gives a past intermediate measurement, representing a fractional delay. For example, if $\alpha_{g,d} = 1$ for $g = \tilde{g}$ and $d = \tilde{d}$ and $\alpha_{g,d} = 0$ for all other g and d, then $\mathbf{y}_k = \mathbf{z}_{\tilde{g},\tilde{d}}$.

For the simulation purpose, we define p_d as the delay probability. We distribute p_d equally across all past intermediate instants within the maximum delay M_d. We also get $1-p_d$ as the probability of no delay, giving the probability of $\mathbf{y}_k = \mathbf{z}_k$. These probabilities are used for generating $\alpha_{g,d}$. We restrict the performance analysis for the maximum delay being up to five sampling instants. Moreover, we consider five intermediate instants between two consecutive sampling in-

stants.

4.4.2 Problem 1

Recalling Chapter 3, the state space model for this problem is as follows:

$$\mathbf{x}_k = a_m \mathbf{x}_{k-1} + b_m \frac{\mathbf{x}_{k-1}}{1+\mathbf{x}_{k-1}^2} + c_m \cos(1.2k\tau) + \nu_k, \tag{4.36}$$

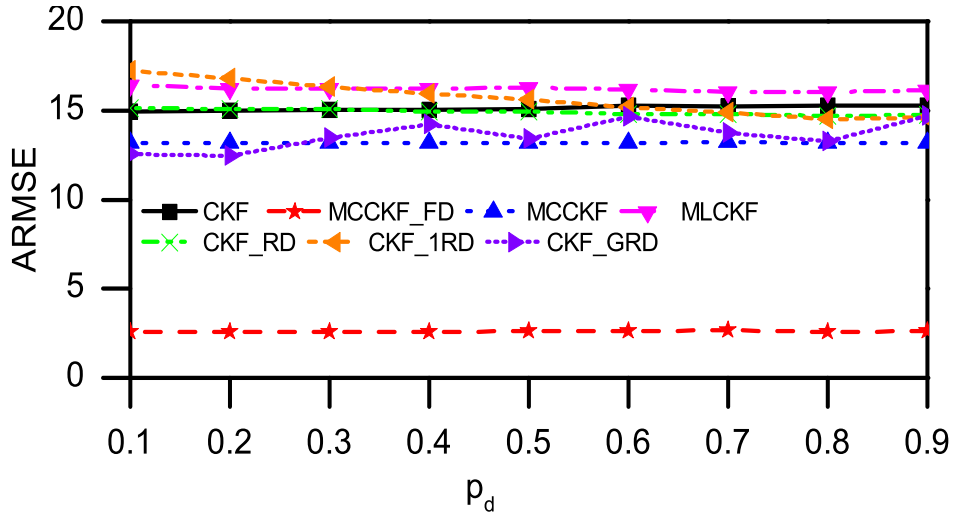
$$\mathbf{z}_k = \mathbf{x}_k^2/d_m + \eta_k. \tag{4.37}$$

Please follow Chapter 3 for more details. For the simulation purpose, we assign $a_m = 5$, $b_m = 25$, $c_m = 8$, $d_m = 20$, $\tau = 1$, $\mathbf{x}_0 = 0.1$, $\mathbf{Q}_k = 0.3$, and $\tau = 1$. The initial estimate is taken as $\hat{\mathbf{x}}_0 = 9\mathscr{N}(\mathbf{x}_0, \mathbf{P}_{0|0})$, with initial covariance $\mathbf{P}_{0|0} = 5$. The simulation is performed for $T_S = 200$ time-steps and 500 Monte-Carlo runs. The non-Gaussian measurement noise is taken as $\eta_k = 0.9\mathscr{N}(0, \mathbf{R}1) + 0.1\mathscr{N}(0, \mathbf{R}2)$, with $\mathbf{R}1 = 1$ and $\mathbf{R}2 = 1000$.

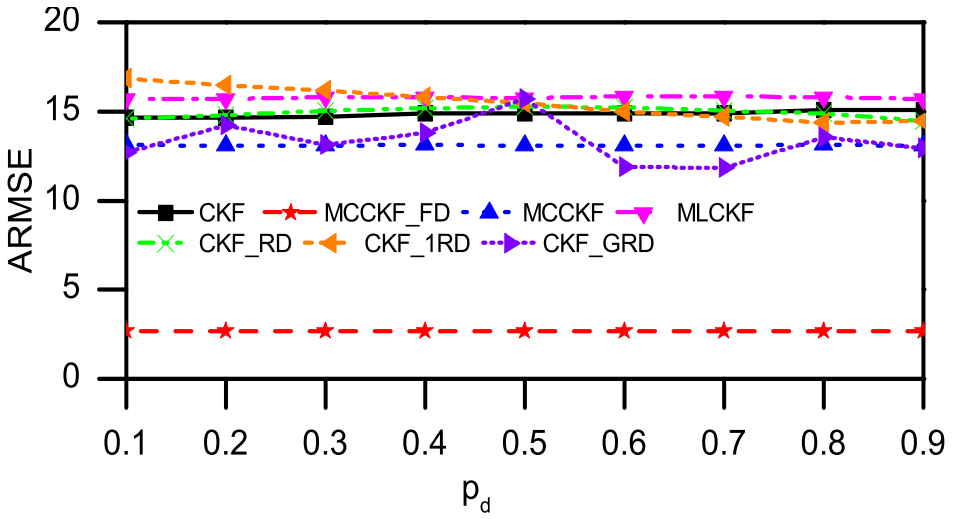
Fig. 4.3 plots the ARMSEs with varying delay probability p_d for $M_d = 1$ and $M_d = 3$. Moreover, Table 4.1 provides the ARMSEs for two distinct delay probabilities ($p_d = 0.2$ and $p_d = 0.5$) by varying M_d from one to five. The figure and table collectively infer a reduced ARMSE of the MCCKF_FD, concluding an improved accuracy of the GFMCFD. Furthermore, we compare the relative computational times of all filters in Table 4.4, which concludes a relatively increased computational time of the MCCKF_FD (in general, the GFMCFD).

Table 4.1: ARMSE comparison of the proposed and existing filters for varying maximum delay M_d under two different delay probabilities (0.2 and 0.5).

p_d	M_d	CKF	MCCKF_FD	MCCKF	MLCKF	CKF_RD	CKF_1RD	CKF_GRD
0.2	1	14.9653	2.5855	13.1550	16.2072	15.0877	16.7849	12.4356
	2	14.7067	2.7784	13.1353	16.4587	15.4439	16.6552	13.3541
	3	14.6715	2.6551	13.0961	15.7229	14.7851	16.4592	14.2315
	4	14.6053	2.7213	13.1334	15.0967	14.5129	16.3589	12.8144
	5	14.6764	2.9240	13.1108	15.0757	14.3007	16.4281	12.8774
0.5	1	14.6317	2.6652	13.1250	15.708	14.6013	16.8475	12.6875
	2	15.0122	2.7703	13.1568	16.3016	15.0822	15.4912	14.0018
	3	14.8951	2.6592	13.0631	15.7674	15.2515	15.4471	15.7139
	4	14.8988	2.7235	13.1141	15.297	15.0275	15.4195	13.7014
	5	14.7828	2.9262	13.0868	15.0895	14.9574	15.4476	14.3854



(a)



(b)

Figure 4.3: ARMSE plots of the proposed and existing filters for varying p_d, with (a) $M_d = 1$ and (b) $M_d = 3$.

4.4.3 Problem 2

This problem is the same as Problem 2 of Chapter 3, which is represented by the following state space model:

$$\mathbf{x}_k = I_6 \mathbf{x}_{k-1} + \nu_k. \tag{4.38}$$

$$\mathbf{z}_k = \left[\sum_{j=1}^{3} a_{j,k} \cos(2\pi f_{j,k} k\tau) \quad \sum_{j=1}^{3} a_{j,k} \sin(2\pi f_{j,k} k\tau) \right]^T + \eta_k, \tag{4.39}$$

For true data simulation, we assign $\tau = 1.667 \times 10^{-6}$ second. The process noise covariance is taken as $\mathbf{Q} = diag([\sigma_f^2\ \sigma_f^2\ \sigma_f^2\ \sigma_a^2\ \sigma_a^2\ \sigma_a^2])$ with $\sigma_f = 22.5\ mHz$ and $\sigma_a = 20\ mV$, while we take the initial

state as $\mathbf{x}_0 = [1000, 200, 1800, 2, 5, 3]^T$. The initial estimate is considered as $\hat{\mathbf{x}}_{0|0} = \mathcal{N}(\mathbf{x}_0, \mathbf{P}_{0|0})$, with $\mathbf{P}_{0|0} = 10^{-3} \times diag([20, 50, 100, 5, 4, 2])$. The filtering is performed over $T_S = 800$ time-steps and the ARMSE is computed by implementing 500 Monte-Carlo simulations. The non-Gaussian measurement noise is taken as $\eta_k = 0.9\mathcal{N}(0, \mathbf{R}) + 0.1\mathcal{N}(0, 1000\mathbf{R})$ with $\mathbf{R} = diag([0.9 \;\; 0.9])$.

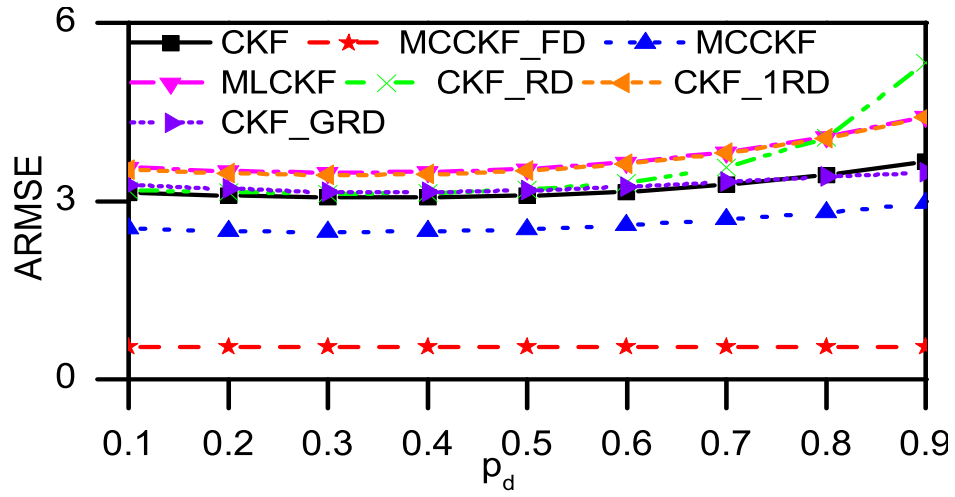
In this problem, we compute the ARMSE as the mean of the ARMSEs of the three sinusoids. Subsequently, we plot the ARMSEs of frequency and amplitude against varying p_d in Fig. 4.4 and 4.5 for $M_d = 1$ and $M_d = 3$, respectively. Moreover, we show the ARMSEs for various M_d in Tables 4.2 and 4.3 for frequency and amplitude, respectively, considering $p_d = 0.2$ and $p_d = 0.5$. We further analyze the filtering performance by varying the initial covariance, indicating a diversified initial guess of the sinusoids. In this regard, we consider three different cases, to be referred as Case 1, Case 2, and Case 3, with initial covariances $10\mathbf{P}_{0|0}$, $20\mathbf{P}_{0|0}$, and $50\mathbf{P}_{0|0}$, respectively. Conversely, we compare the ARMSEs of the proposed and existing filters in Fig. 4.6 for $p_d = 0.5$ and $M_d = 3$. The figures and tables conclude a reduced estimation error of the MCCKF_FD, which further concludes an improved accuracy of the GFMCFD. The computational times of all filters followed a similar trend as Table 4.4, concluding the relatively increased computational time of the GFMCFD.

Table 4.2: Frequency ARMSE comparison of the proposed and existing filters for varying maximum delay M_d under two different delay probabilities (0.2 and 0.5).

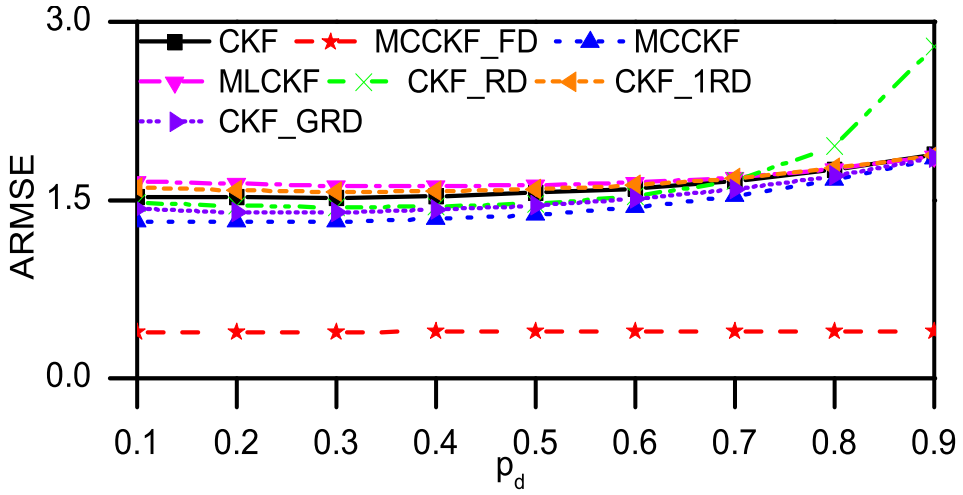
p_d	M_d	CKF	MCCKF_FD	MCCKF	MLCKF	CKF_RD	CKF_1RD	CKF_GRD
	1	3.0928	0.5404	2.4975	3.5094	3.1452	3.4653	3.2084
	2	3.2697	0.5534	2.6221	3.7668	3.3161	3.7161	3.3808
0.2	3	3.4633	0.5445	2.7917	3.9986	3.5137	3.9475	3.5927
	4	3.6365	0.5432	2.9305	4.2113	3.7044	4.1637	3.8194
	5	3.8463	0.5486	3.1008	4.4510	3.9707	4.4100	4.1168
	1	3.0947	0.5403	2.5166	3.5413	3.1908	3.5078	3.1759
	2	3.3087	0.5536	2.6776	3.8620	3.4313	3.8240	3.3913
0.5	3	3.5564	0.5445	2.8741	4.1806	3.7160	4.1386	3.6306
	4	3.7922	0.5433	3.0468	4.4726	4.0808	4.4320	3.9117
	5	3.9807	0.5481	3.1880	4.6821	4.5677	4.6554	4.2850

4.4.4 Analysis and Discussion

In this subsection, we further illustrate the results and extract some interesting conclusions. We observe, in Figs. 4.3 to 4.4, that the ARMSE is increasing (accuracy is deteriorating) with



(a)

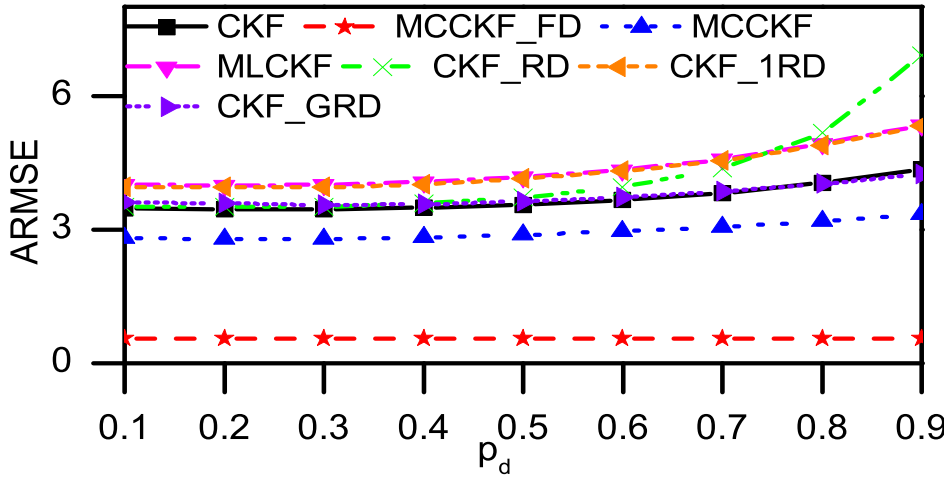


(b)

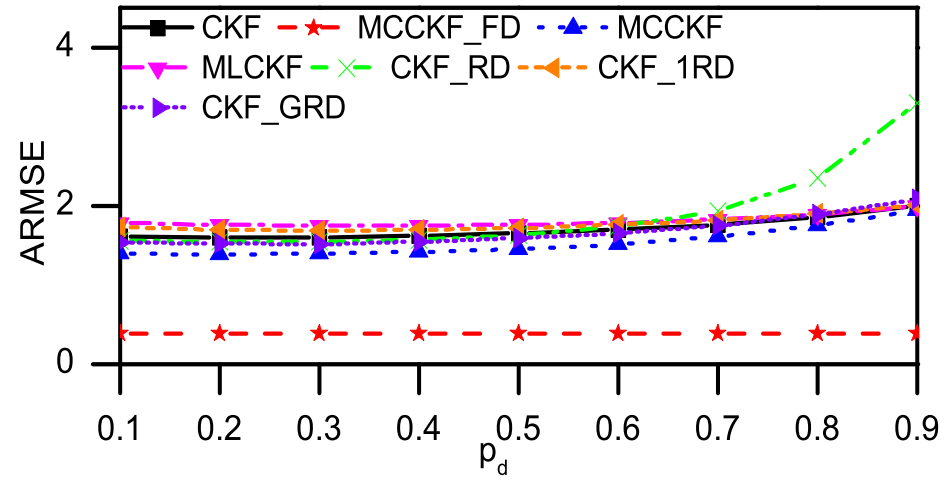
Figure 4.4: ARMSE plots of the proposed and existing filters for varying p_d, with M_d=1: a) frequency and b) amplitude.

Table 4.3: Amplitude ARMSE comparison of the proposed and existing filters for varying maximum delay M_d under two different delay probabilities (0.2 and 0.5).

p_d	M_d	CKF	MCCKF_FD	MCCKF	MLCKF	CKF_RD	CKF_1RD	CKF_GRD
	1	1.5236	0.3896	1.3150	1.6415	1.4518	1.5834	1.3981
	2	1.5888	0.3902	1.3739	1.7181	1.5197	1.6566	1.4759
0.2	3	1.6011	0.3919	1.3852	1.7647	1.5428	1.6956	1.5210
	4	1.6423	0.3847	1.4214	1.8314	1.5910	1.7549	1.5963
	5	1.6729	0.3839	1.4507	1.8824	1.6507	1.8025	1.6960
	1	1.5575	0.3914	1.3756	1.6241	1.4705	1.5887	1.4532
	2	1.6326	0.3917	1.4371	1.7120	1.5642	1.6759	1.5382
0.5	3	1.6526	0.3918	1.4532	1.7582	1.6290	1.7173	1.5874
	4	1.6957	0.3835	1.4863	1.8257	1.7403	1.7798	1.6620
	5	1.7169	0.3825	1.5048	1.8831	1.9281	1.8256	1.7817



(a)



(b)

Figure 4.5: ARMSE plots of the proposed and existing filters for varying p_d, with M_d=3: a) frequency and b) amplitude.

Table 4.4: Relative computational times of the proposed and the existing filters.

Filter	Relative computational time
CKF	1
MCCKF_FD	8.36
MCCKF	6.63
MLCKF	3.59
CKF_RD	1.58
CKF_1RD	1.26
CKF_GRD	1.70

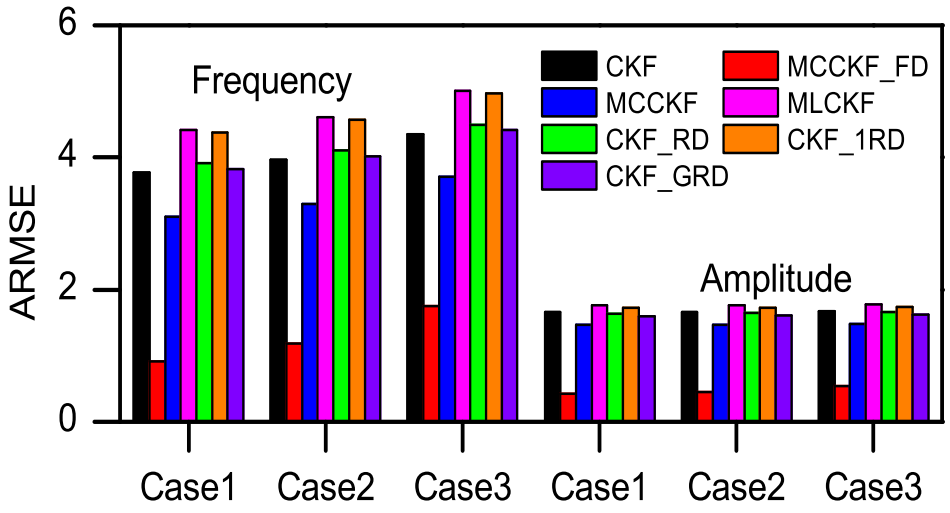


Figure 4.6: ARMSEs of amplitude and frequency for three different cases (initial covariances).

the increasing delay probability p_d, which a well expected. At the same time, we observe that the ARMSE variation is not very large for p_d being in the range of 0.1 to 0.5. This concludes that the filtering performance is not significantly harmed if only a few measurements are delayed, while others are time-synchronized. A similar pattern is observed in [99] and [159] as well. Nevertheless, we may have a different trend as the system dynamics and environment change. Furthermore, unlike the existing filters, the GFMCFD is not much affected by even a further increase in p_d. This concludes that GFMCFD delay identification is accurately identifying the delay and, accordingly, the GFMCFD is efficiently tracking the state dynamics.

In Table 4.1, unexpectedly, with the increasing maximum delay M_d, the ARMSE decreases (accuracy improves) for the MLCKF, CKF_RD, and CKF_GRD. Such a response is although unusual, interestingly, in this case, it may be reasonable. This is because the MLCKF, CKF_RD, and CKF_GRD are extensions of the CKF under the minimum mean square error (MMSE) criterion, where the noises are assumed as Gaussian. Please note that, even though they are designed for handling the delayed measurements, they can not extract the information lost due to the delay exactly. Thus, the partially lost information due to the delay can be treated as noises. Therefore, in the case of increasing M_d, it is possible that the increased loss of information may be causing a closer approximation of the net noise as Gaussian and their filtering accuracy may be improving. This must not, however, be the case always. Therefore, in other cases, so in other tables (Tables 4.2 and 4.3), we observe a different pattern, and their ARMSE increases (accuracy deteriorates) as M_d increases.

4.5 Summary

The existing Gaussian filtering traditionally ignores various measurement irregularities, restraining the development and integration of several state-of-art technologies. For example, ignoring measurement delay often prevents the integration of transmission channels, while ignoring non-Gaussian measurement noises commonly precludes the development of high-precision technologies. The GFMCFD addresses the delayed measurements and non-Gaussian measurement noises jointly to mitigate this problem. The GFMCFD is designed under the MC criterion to rationalize the higher-order moments of noises and address the non-Gaussianity. Moreover, it stochastically identifies the delays and restructures the filtering strategy accordingly, to address the delay. Interestingly, the GFMCFD addresses fractional delays, where even the Gaussian filtering extensions, primarily designed for handling the integer delays, mostly underperform. Furthermore, the GFMCFD applies to any of the traditional Gaussian filters, such as the CKF and GHF. The performance analysis reveals an improved accuracy of the GFMCFD. Although this development is capable of addressing two irregularities at once, it is worth noting that the literature reports more irregularities that can be attributed solely to cyber-attacks.

Chapter 5

Gaussian Filtering with Cyber-Attacked Data

5.1 Introduction

In Chapter 2, we discussed about cyber-attacks on measurement data and its different types, particularly, the FDI, TAM, and DoS attacks. We also reviewed the various developments in nonlinear filtering for handling the FDI, TAM, and DoS attacks individually. However, the reviewed filtering methods are incapable of handling the simultaneously occurring FDI, TAM, and DoS attacks. Consequently, the existing methods for handling cyber-attacks underperform or fail if one or more of the FDI, TAM, and DoS attacks occur simultaneously.

In this chapter, we introduce a new extension of Gaussian filtering for handling the simultaneously occurring FDI, TAM, and DoS attacks. We name the proposed method as Gaussian filtering under FDI, TAM, and DoS attacks, which will be abbreviated as GFFTD. To develop the GFFTD, we reformulate the traditional measurement model for incorporating the possibility of the simultaneously occurring FDI, TAM, and DoS attacks. Subsequently, we re-derive the traditional Gaussian filtering method for the reformulated measurement model. Interestingly, the GFFTD is applicable to any of the existing Gaussian filters, such as the UKF, CKF, and GHF. We validate the improved accuracy of the GFFTD for two simulation problems.

5.2 Problem Formulation

Following the notations and discussions in Chapter 1, we consider the following state space model for representing a general nonlinear dynamical system:

$$\mathbf{x}_k = \phi_{k-1}(\mathbf{x}_{k-1}) + \nu_{k-1}, \tag{5.1}$$

$$\mathbf{z}_k = \Psi_k(\mathbf{x}_k) + \eta_k. \tag{5.2}$$

In this problem, $\mathbf{z}_k$ is tampered due to cyber-attack, giving a tampered measurement $\mathbf{y}_k$. This chapter aims to develop an advanced Gaussian filtering method for estimating $\mathbf{x}_k$ from $\mathbf{y}_k$.

To keep the attack masked, the intruders generally tamper the data intermittently. The author assumes that, at any particular instant, the data is tampered by only one of the FDI, TAM, and DoS attacks. We model the attacks by reformulating the measurement model (5.2) for $\mathbf{y}_k$, considering the following modeling strategies.

- We utilize three Bernoulli random variables, α_k, β_k, and γ_k, to characterize the occurrences of no-attack and different forms of attack. This characterization may include the following steps: i) A1: likelihood computation to recognize the no-attack and characterize α_k, ii) A2: correlation analysis to recognize the delay (TAM-attack) and characterize γ_k, iii) A3: a data arrival with true in A1 and false in A2 inherently infer the FDI attack, characterizing β_k, and iv) A4: no data arrival naturally infers the DoS attack.

- The FDI attack can be compensated by approximating the uncertain false data with a Gaussian distribution. For this purpose, we can identify a suitable Gaussian distribution from data preprocessing (after the FDI attack is detected). Following [138], the data preprocessing may include likelihood analysis, predefined heuristic rules, and normalizations.

- The TMA attack, causing the delay, can be compensated by incorporating the possibility of receiving past measurements. In this regard, we utilize the Geometric distribution for modeling the instantaneous delays.

Please note that the data pre-analysis and preprocessing are performed beforehand and they are not part of our development.

In the remaining parts of this chapter, for any random variable ϕ and any probability p, we use the notations ϕ' and p', respectively, for denoting $(1-\phi)$ and $(1-p)$, respectively. This applies to all random variables and probabilities.

The Bernaoulli random variables α_k, β_k, and γ_k obey

$$\begin{cases} P(\alpha_k = 1) = \mathbb{E}[\alpha_k] = \mathbb{E}[(\alpha_k)^m] = p_n \\ P(\beta_k = 1) = \mathbb{E}[\beta_k] = \mathbb{E}[(\beta_k)^m] = p_f \\ P(\gamma_k = 1) = \mathbb{E}[\gamma_k] = \mathbb{E}[(\gamma_k)^m] = p_d \\ P(\alpha_k = 0) = \mathbb{E}\left[\alpha'_k\right] = \mathbb{E}\left[(\alpha'_k)^m\right] = p'_n \\ P(\beta_k = 0) = \mathbb{E}\left[\beta'_k\right] = \mathbb{E}\left[(\beta'_k)^m\right] = p'_f \\ P(\gamma_k = 0) = \mathbb{E}\left[\gamma'_k\right] = \mathbb{E}\left[(\gamma'_k)^m\right] = p'_d, \end{cases} \tag{5.3}$$

where $\mathbb{E}[\cdot]$ denotes the statistical expectation operator and $m \in \mathbb{R}$ is a constant. Moreover, p_n, p_f, and p_d denote the probabilities of FDI, TAM, and DoS attacks, respectively.

Similar to Eq. (5.3), the other statistical parameters corresponding to α_k, β_k, and γ_k are determined as

$$\begin{cases} \mathbb{E}\left[(\alpha_k - p_n)^2\right] = \mathbb{E}\left[(\alpha_k)^2\right] - \mathbb{E}[\alpha_k]^2 = p_n p'_n \\ \mathbb{E}\left[(\beta_k - p_f)^2\right] = \mathbb{E}\left[(\beta_k)^2\right] - \mathbb{E}[\beta_k]^2 = p_f p'_f \\ \mathbb{E}\left[(\gamma_k - p_d)^2\right] = \mathbb{E}\left[(\gamma_k)^2\right] - \mathbb{E}[\gamma_k]^2 = p_d p'_d \end{cases} \tag{5.4}$$

Furthermore, let us denote $G_{d,k}$ as a geometric random variable, characterizing d-delay (delays up to d sampling intervals) at t_k and p_g as the probability of getting a unity value for each entry of $G_{d,k}$. Then,

$$\begin{cases} P(G_{d,k}(i) = 1) = \mathbb{E}[(G_{d,k}(i))^m] = \Gamma_i \\ P(G_{d,k}(i) = 0) = \mathbb{E}[(G_{d,k}(i)')^m] = \Gamma'_i \\ \mathbb{E}\left[\left(G_{d,k}(i) - \Gamma_i\right)^2\right] = \Gamma_i \Gamma'_i, \end{cases} \tag{5.5}$$

where $\Gamma_i = (p'_g)^{i-1} p_g$ is the probability of i-delay $\forall i \in \{1, 2, \cdots, d\}$, if TAM attack occurs at t_k.

Let us now denote Δ_k as the false data injection at t_k, if we have an FDI attack at t_k. Then, we approximate $\Delta_k \approx \mathcal{N}(\hat{\delta}, \Sigma_\delta)$, where, as discussed in [138], $\hat{\delta}$ and Σ_δ can be determined from the nature of the predefined heuristic rules and normalization methods.

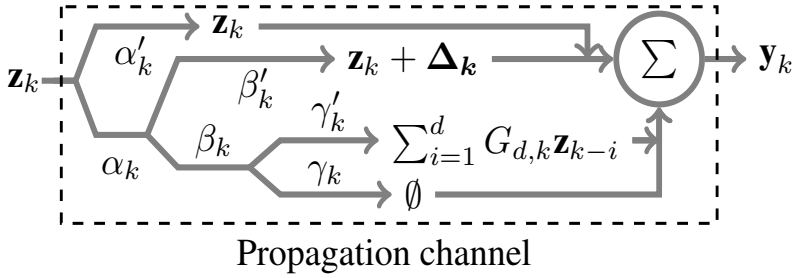


Figure 5.1: Pictorial representation of attack modeling strategy.

Table 5.1: α_k, β_k, and γ_k values for different attacks.

Stochastic parameters	Form of attack
$\alpha_k = 0$, $\beta_k \in \{0,1\}$, $\gamma_k \in \{0,1\}$	No-attack
$\alpha_k = 1$, $\beta_k = 0$, $\gamma_k \in \{0,1\}$	FDI-attack
$\alpha_k = 1$, $\beta_k = 1$, $\gamma_k = 0$	TAM-attack
$\alpha_k = 1$, $\beta_k = 1$, $\gamma_k = 1$	DoS-attack

To this end, following the modeling strategy pictorially presented in Fig. 5.1, we model the received measurement $\mathbf{y}_k$ as $\mathbf{y}_k = \alpha'_k\mathbf{z}_k + \alpha_k\left[\beta'_k(\mathbf{z}_k + \Delta_k) + \beta_k\gamma'_k\sum_{i=1}^{d} G_{d,k}(i)\mathbf{z}_{k-i}\right]$, which can be further expressed as

$$\mathbf{y}_k = (1 - \alpha_k\beta_k)\mathbf{z}_k + \alpha_k\beta'_k\Delta_k + \alpha_k\beta_k\gamma'_k\sum_{i=1}^{d} G_{d,k}(i)\mathbf{z}_{k-i}. \tag{5.6}$$

Table 5.1 demonstrates α_k, β_k, and γ_k values for different attacks.

To this end, the objective of this chapter is to re-derive the traditional Gaussian filtering method for the modified measurement model (5.6).

5.3 Gaussian Filtering under Cyber-Attack

In this section, we derive the GFFTD for filtering with cyber-attacked measurements. As the cyber-attacks influence only the measurements, we derive the GFFTD by re-deriving the traditional Gaussian filtering expressions, consisting of measurement. Interestingly, there are only three such expressions: measurement estimate $\hat{\mathbf{z}}_{k|k-1}$, measurement covariance $\mathbf{P}^{\mathbf{zz}}_{k|k-1}$, and cross-covariance $\mathbf{P}^{\mathbf{xz}}_{k|k-1}$. We re-derive these expressions with respect to $\mathbf{y}_k$, giving $\hat{\mathbf{y}}_{k|k-1}$, $\mathbf{P}^{\mathbf{yy}}_{k|k-1}$ and $\mathbf{P}^{\mathbf{xy}}_{k|k-1}$. Consequently, the GFFTD is obtained by replacing $\hat{\mathbf{z}}_{k|k-1}$, $\mathbf{P}^{\mathbf{zz}}_{k|k-1}$ and $\mathbf{P}^{\mathbf{xz}}_{k|k-1}$ with $\hat{\mathbf{y}}_{k|k-1}$, $\mathbf{P}^{\mathbf{yy}}_{k|k-1}$ and $\mathbf{P}^{\mathbf{xy}}_{k|k-1}$, respectively in the traditional Gaussian filtering.

We derive $\hat{\mathbf{y}}_{k|k-1}$, $\mathbf{P}^{\mathbf{yy}}_{k|k-1}$, and $\mathbf{P}^{\mathbf{xy}}_{k|k-1}$ with respect to $\mathbf{y}_k$ (modeled in Eq. (5.6)) through the three

subsequent theorems.

Theorem 5.3.1 *The estimate of the cyber-attacked measurement $\mathbf{y}_k$, i.e. $\hat{\mathbf{y}}_{k|k-1}$, can be determined as*

$$\hat{\mathbf{y}}_{k|k-1} = (1-p_n p_f)\hat{\mathbf{z}}_{k|k-1} + p_n p'_f \hat{\delta} + p_n p_f p'_d \sum_{i=1}^{d} \Gamma_i \hat{\mathbf{z}}_{k-i|k-1}. \tag{5.7}$$

Proof: As $\hat{\mathbf{y}}_{k|k-1} = \mathbb{E}\left[\mathbf{y}_{k|k-1}\right]$, for $\mathbf{y}_k$ given in Eq. (5.7), we obtain

$$\hat{\mathbf{y}}_{k|k-1} = \mathbb{E}\left[(1-\alpha_k\beta_k)\mathbf{z}_k + \alpha_k\beta'_k\Delta_k + \alpha_k\beta_k\gamma'_k \sum_{i=1}^{d} G_{d,k}(i)\mathbf{z}_{k-i}\right].$$

As α_k, β_k, γ_k, and $G_{d,k}(i)$ characterize the occurrences of the measurement irregularities, they are independent of $\mathbf{z}_k$ and $\mathbf{z}_{k-i}$, depicting the measurement values. Thus, as $\mathbb{E}\left[\mathbf{z}_{k|k-1}\right] = \hat{\mathbf{z}}_{k|k-1}$, we get

$$\hat{\mathbf{y}}_{k|k-1} = \mathbb{E}\left[1-\alpha_k\beta_k\right]\hat{\mathbf{z}}_{k|k-1} + \mathbb{E}\left[\alpha_k\beta'_k\right]\hat{\delta} + \mathbb{E}\left[\alpha_k\beta_k\gamma'_k\right]\sum_{i=1}^{k-1}\mathbb{E}\left[G_{d,k}(i)\right]\hat{\mathbf{z}}_{k-i|k-1}.$$

Substituting $\mathbb{E}[\alpha_k]$, $\mathbb{E}[\beta_k]$, $\mathbb{E}\left[\gamma'_k\right]$, and $\mathbb{E}[G_{d,k}(i)]$ from Eqs. (5.3), (5.4) and (5.5) the above equation reduces to Eq. (5.7). □

Theorem 5.3.2 *The covariance matrix $\boldsymbol{P}^{\boldsymbol{yy}}_{k|k-1}$ for $\mathbf{y}_k$ can be given in the form of Eq.* (5.8)*.*

$$\begin{aligned}\mathbf{P}^{\mathbf{yy}}_{k|k-1} =& (1-p_n p_f)\mathbf{P}^{\mathbf{zz}}_{k|k-1} + (1-p_n p_f)p_n p_f \hat{\mathbf{z}}_{k|k-1}(\hat{\mathbf{z}}_{k|k-1})^T + p'_f p_n \Sigma_\delta + (p_n p'_f - p_n^2 p'^2_f) \\ &\times \hat{\delta}\hat{\delta}^T + p_n p_f p'_d \sum_{i=1}^{d} \Gamma_i \mathbf{P}^{\mathbf{zz}}_{k-i|k-1} + \sum_{i=1}^{d} \left(p_n p_f p'_d \Gamma_i (1-p_n p_f p'_d \Gamma_i)\right) \hat{\mathbf{z}}_{k-i|k-1}\hat{\mathbf{z}}^T_{k-i|k-1} \\ &+ \sum_{i\neq j=1}^{d} \left(p_n p_f p'_d (p'_g)^{i+j-2} p_g^2 (1-p_n p_f p'_d (p'_g)^{i+j-2} p_g^2)\right) \hat{\mathbf{z}}_{k-i|k-1}\hat{\mathbf{z}}^T_{k-j|k-1}.\end{aligned} \tag{5.8}$$

Proof: The covariance matrix $\mathbf{P}^{\mathbf{yy}}_{k|k-1}$ is given as

$$\mathbf{P}^{\mathbf{yy}}_{k|k-1} = \mathbb{E}\left[(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1})(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1})^T\right]. \tag{5.9}$$

For $\mathbf{y}_k$ and $\hat{\mathbf{y}}_{k|k-1}$ given in Eqs. (5.6) and (5.7), respectively, we obtain $\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}$ in the form of Eq. (5.10). Then, substituting $\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}$ from Eq. (5.10) into Eq. (5.11), we obtain $\mathbf{P}^{\mathbf{yy}}_{k|k-1} =$

$$\begin{aligned}\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1} = &\underbrace{(1-\alpha_k\beta_k)(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})}_{\mathbb{M}_1} + \underbrace{\alpha_k\beta_k'(\Delta_k - \hat{\delta})}_{\mathbb{M}_2} + \underbrace{(p_n p_f - \alpha_k\beta_k)\hat{\mathbf{z}}_{k|k-1}}_{\mathbb{M}_3} \\ &+ \underbrace{(\alpha_k\beta_k' - p_n p_f')\hat{\delta}}_{\mathbb{M}_4} + \underbrace{\alpha_k\beta_k\gamma_k' \sum_{i=1}^{d} G_{d,k}(i)(\mathbf{z}_{k-i} - \hat{\mathbf{z}}_{k-i|k-1})}_{\mathbb{M}_5} \\ &+ \underbrace{\sum_{i=1}^{d} \left(\alpha_k\beta_k\gamma_k' G_{d,k}(i) - p_n p_f p_d' \Gamma_i\right) \hat{\mathbf{z}}_{k-i|k-1}}_{\mathbb{M}_6}.\end{aligned} \tag{5.10}$$

$\sum_{i=1}^{6}\sum_{j=1}^{6} \mathbb{E}\left[\mathbb{M}_i\mathbb{M}_j^T\right]$. Using stochastic independence properties for independent random variables, we can trivially conclude $\mathbb{E}[\mathbb{M}_i\mathbb{M}_j^T] = 0 \; \forall \; i \neq j$, giving

$$\mathbf{P}^{\mathbf{yy}}_{k|k-1} = \sum_{i=1}^{6} \mathbb{E}\left[\mathbb{M}_i\mathbb{M}_i^T\right]. \tag{5.11}$$

We now derive $\mathbb{E}[\mathbb{M}_i\mathbb{M}_i^T] \; \forall i \in \{1,2,\cdots,6\}$, which we add later to obtain $\mathbf{P}^{\mathbf{yy}}_{k|k-1}$.

For $\mathbb{M}_1$ given in Eq. (5.10), we can write

$$\mathbb{E}\left[\mathbb{M}_1\mathbb{M}_1^T\right] = \mathbb{E}\left[(1-\alpha_k\beta_k)^2(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})^T\right].$$

Please note that α_k and β_k are independent of $\mathbf{z}_k$ and $\hat{\mathbf{z}}_{k|k-1}$. Moreover, as α_k and β_k are independent Bernoulli random variables, we obtain $\mathbb{E}\left[(1-\alpha_k\beta_k)^2\right] = (1-p_n p_f)$. Following this, the above equation is simplified as

$$\mathbb{E}\left[\mathbb{M}_1\mathbb{M}_1^T\right] = (1-p_n p_f)P^{\mathbf{zz}}_{k|k-1}. \tag{5.12}$$

Similarly, for $\mathbb{M}_2$ given in Eq. (5.10), we obtain

$$\mathbb{E}\left[\mathbb{M}_2\mathbb{M}_2^T\right] = \mathbb{E}\left[\alpha_k^2\beta_k'^2\right]\mathbb{E}\left[(\Delta_k - \hat{\delta})(\Delta_k - \hat{\delta})^T\right].$$

As α_k and β_k are independent, we get

$$\mathbb{E}\left[\mathbb{M}_2\mathbb{M}_2^T\right] = p_n p_f' \Sigma_\delta. \tag{5.13}$$

Substituting $\mathbb{M}_3$ from Eq. (5.10) into $\mathbb{E}\left[\mathbb{M}_3\mathbb{M}_3^T\right]$, and simplifying further, we get

$$\mathbb{E}\left[\mathbb{M}_3\mathbb{M}_3^T\right] = p_n p_f (1 - p_n p_f)\hat{\mathbf{z}}_{k|k-1}\hat{\mathbf{z}}_{k|k-1}^T. \tag{5.14}$$

Substituting $\mathbb{M}_4$ from Eq. (5.10), we obtain $\mathbb{E}\left[\mathbb{M}_4\mathbb{M}_4^T\right]$ as

$$\mathbb{E}\left[\mathbb{M}_4\mathbb{M}_4^T\right] = \mathbb{E}\Big[(\alpha_k\beta_k' - p_n p_f')^2\Big]\mathbb{E}\Big[\hat{\delta}\hat{\delta}^T\Big],$$

which is further simplified as

$$\mathbb{E}\left[\mathbb{M}_4\mathbb{M}_4^T\right] = (p_n p_f' - p_n^2 p_f'^2)\hat{\delta}\hat{\delta}^T. \tag{5.15}$$

We now substitute $\mathbb{M}_5$ from Eq. (5.10) into $\mathbb{E}\left[\mathbb{M}_5\mathbb{M}_5^T\right]$ to obtain

$$\mathbb{E}\left[\mathbb{M}_5\mathbb{M}_5^T\right] = \mathbb{E}\Big[\alpha_k\beta_k\gamma_k' \sum_{i=1}^{d} G_{d,k}(i)(\mathbf{z}_{k-i} - \hat{\mathbf{z}}_{k-i|k-1})\alpha_k\beta_k\gamma_k' \sum_{j=1}^{d} G_{d,k}(j)(\mathbf{z}_{k-j} - \hat{\mathbf{z}}_{k-j|k-1})^T\Big].$$

Let us consider the following notes: i) $\mathbf{z}_{k-i}$ and $\mathbf{z}_{k-j}$ are independent $\forall i \neq j$, ii) $G_{d,k}(i)$ and $G_{d,k}(j)$ are independent $\forall i \neq j$, and iii) α_k, β_k, and γ_k are independent of each other, and also, they are independent of $\mathbf{z}_{k-i}$ and $G_{d,k}(i)$, $\forall i \in \{1,2,\cdots,d\}$. Applying their independent properties and simplifying the above equation further, we obtain

$$\mathbb{E}\left[\mathbb{M}_5\mathbb{M}_5^T\right] = p_n p_f p_d' \sum_{i=1}^{d} \Gamma_i \mathbf{P}_{k-i|k-1}^{\mathbf{zz}}. \tag{5.16}$$

Finally, for $\mathbb{M}_6$ given in Eq. (5.10), we obtained as

$$\begin{aligned}\mathbb{E}\left[\mathbb{M}_6\mathbb{M}_6^T\right] = &\sum_{i=1}^{d} \mathbb{E}\left[\left(\alpha_k\beta_k\gamma_k' G_{d,k}(i) - p_n p_f p_d' \Gamma_i\right)^2\right]\mathbb{E}\left[\hat{\mathbf{z}}_{k-i|k-1}\hat{\mathbf{z}}_{k-i|k-1}^T\right] + \sum_{i\neq j=1}^{d} \mathbb{E}\Big[\left(\alpha_k\beta_k\gamma_k'\right. \\ &\left. G_{d,k}(i) - p_n p_f p_d'\Gamma_i\right)\left(\alpha_k\beta_k\gamma_k' G_{d,k}(j) - p_n p_f p_d'\Gamma_j\right)\Big]\mathbb{E}\left[\hat{\mathbf{z}}_{k-i|k-1}\hat{\mathbf{z}}_{k-j|k-1}^T\right].\end{aligned} \tag{5.17}$$

Further, applying various independence properties for various independent random variables,

then Eq. (5.17) can be expressed as

$$\mathbb{E}\left[\mathbb{M}_6\mathbb{M}_6^T\right] = \sum_{i=j=1}^{d} p_n p_f p_d' \Gamma_i (1 - p_n p_f p_d' \Gamma_i) \hat{\mathbf{z}}_{k-i|k-1} \hat{\mathbf{z}}_{k-i|k-1}^T + \sum_{i\neq j=1}^{d} \left(p_n p_f p_d' (p_g')^{i+j-2} p_g^2\right) \left(1 - \left(p_n p_f p_d' (p_g')^{i+j-2} p_g^2\right)\right) \hat{\mathbf{z}}_{k-i|k-1} \hat{\mathbf{z}}_{k-j|k-1}^T. \tag{5.18}$$

Substituting $\mathbb{E}[\mathbb{M}_1\mathbb{M}_1^T]$, $\mathbb{E}[\mathbb{M}_2\mathbb{M}_2^T]$, $\mathbb{E}[\mathbb{M}_3\mathbb{M}_3^T]$, $\mathbb{E}[\mathbb{M}_4\mathbb{M}_4^T]$, $\mathbb{E}[\mathbb{M}_5\mathbb{M}_5^T]$, and $\mathbb{E}[\mathbb{M}_6\mathbb{M}_6^T]$ from Eqs. (5.12), (5.13), (5.14), (5.15), (5.16), and (5.18), respectively, into Eq. (5.11), $\mathbf{P}_{k|k-1}^{\mathbf{yy}}$ can be expressed in the form of Eq. (5.9). □

Theorem 5.3.3 *The cross-covariance matrix between $\boldsymbol{x}_k$ and $\mathbf{y}_k$ can be obtained as*

$$\boldsymbol{P}_{k|k-1}^{\boldsymbol{xy}} = (1 - p_n p_f) \boldsymbol{P}_{k|k-1}^{\boldsymbol{xz}} + p_n p_f p_d' \sum_{i=1}^{d} \Gamma_i \boldsymbol{P}_{k-i|k-1}^{\boldsymbol{xz}}. \tag{5.19}$$

Proof: For $\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}$ given in Eq. (5.10), we get $\mathbf{P}_{k|k-1}^{\mathbf{xy}} = \sum_{i=1}^{6} \mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})\mathbb{M}_i^T\right]$. As $\hat{\mathbf{z}}_{k|k-1}$, $\hat{\mathbf{z}}_{k-i|k-1}$, and $\hat{\delta}_k$ are constants and $\mathbf{x}_k$ is independent of Δ_k, we can conclude that $\sum_l \mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})\mathbb{M}_i^T\right] = 0$, $\forall i \in \{2,3,4,6\}$, giving

$$\mathbf{P}_{k|k-1}^{\mathbf{xy}} = \mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})\mathbb{M}_1^T\right] + \mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})\mathbb{M}_5^T\right]. \tag{5.20}$$

To this end, for $\mathbb{M}_1$ given in Eq. (5.10), we obtain

$$\mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})\mathbb{M}_1^T\right] = \mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(1 - \alpha_k\beta_k)(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})^T\right]$$

which is simplified as

$$\mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})\mathbb{M}_1^T\right] = (1 - p_n p_f) P_{k|k-1}^{\mathbf{xz}}. \tag{5.21}$$

Moreover, for $\mathbb{M}_5$ given in Eq. (5.10), we get

$$\mathbb{E}\left[\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}\right)\mathbb{M}_5^T\right] = \mathbb{E}\left[\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}\right)\left(\alpha_k\beta_k\gamma_k' \sum_{i=1}^{d} G_{d,k}(i)(\mathbf{z}_{k-i} - \hat{\mathbf{z}}_{k-i|k-1})^T\right)\right].$$

Applying the independent property, we get the following after a few simplifications and rearrange-

ments.

$$\mathbb{E}\left[(\mathbf{x}_k-\hat{\mathbf{x}}_{k|k-1})\mathbb{M}_5^T\right]=\mathbb{E}\left[\alpha_k\beta_k\gamma_k'\right]\sum_{i=1}^{d}\Big(\mathbb{E}\left[G_{d,k}(i)\right]\mathbb{E}\left[(\mathbf{x}_k-\hat{\mathbf{x}}_{k|k-1})(\mathbf{z}_{k-i}-\hat{\mathbf{z}}_{k-i|k-1})^T\right]\Big).$$

As $\mathbb{E}\left[\alpha_k\beta_k\gamma_k'\right]=p_np_fp_d'$ and $\mathbb{E}\left[G_{d,k}(i)\right]=\Gamma_i$, we obtain

$$\mathbb{E}\left[(\mathbf{x}_k-\hat{\mathbf{x}}_{k|k-1})\mathbb{M}_5^T\right]=p_np_fp_d'\sum_{i=1}^{d}\Gamma_i\mathbf{P}^{\mathbf{xz}}_{k-i|k-1}. \tag{5.22}$$

Substituting Eqs. (5.21) and (5.22) into Eq. (5.20), we get $\mathbf{P}^{\mathbf{xy}}_{k|k-1}$ in the form of Eq. (5.19).□

Recalling previous discussions, the GFFTD is designed by replacing $\hat{\mathbf{z}}_{k|k-1}$, $\mathbf{P}^{\mathbf{zz}}_{k|k-1}$ and $\mathbf{P}^{\mathbf{xz}}_{k|k-1}$ with $\hat{\mathbf{y}}_{k|k-1}$, $\mathbf{P}^{\mathbf{yy}}_{k|k-1}$ and $\mathbf{P}^{\mathbf{xy}}_{k|k-1}$, respectively, which are derived through the three theorems. As $\hat{\mathbf{y}}_{k|k-1}$, $\mathbf{P}^{\mathbf{yy}}_{k|k-1}$ and $\mathbf{P}^{\mathbf{xy}}_{k|k-1}$ account for cyber-attack, the GFFTD addresses the cyber-attacked measurements.

Remark 5.3.1 *The GFFTD utilizes some estimate and covariance expressions from past instants, which increases its storage requirement.*

5.4 Simulation and Results

In this section, we simulate the CKF-based formulation of the GFFTD for two nonlinear filtering problems. The benchmark filters for comparison include: i) [48]: ordinary CKF, ii) [138]: handling the FDI attack, iii) [95], [99], and [159]: handling the delayed measurements, depicting the TAM attack, and iv) [106]: handling the missing measurements, depicting DoS attack. We consider the CKF-based formulations of [138], [95], [99], and [159], abbreviated as CKF_FA, CKF_2D, MLCKF, and CKF_GRD, respectively. Moreover, we consider EKF-based formulation of [106], abbreviated as EKF_M, as [106] is inapplicable for the CKF. The performance metric is considered as root mean square error (RMSE).

Throughout the simulation, we assign $\Delta_k=\mathcal{N}(1,4)$. We also consider three different cases with varying no-attack probabilities, as Case 1: $p_n=0.8$, Case 2: $p_n=0.5$, and Case 3: $p_n=0.2$. In the case of attack, we consider $p_f=p_d=(1-p_n)/3$, giving the probability of DoS as $(1-p_n)/3$. Moreover, we assign $p_g=0.2$, $p_g=0.8$, and $p_g=0.1$ in the respective three cases to characterize the instantaneous delays.

Problem 1

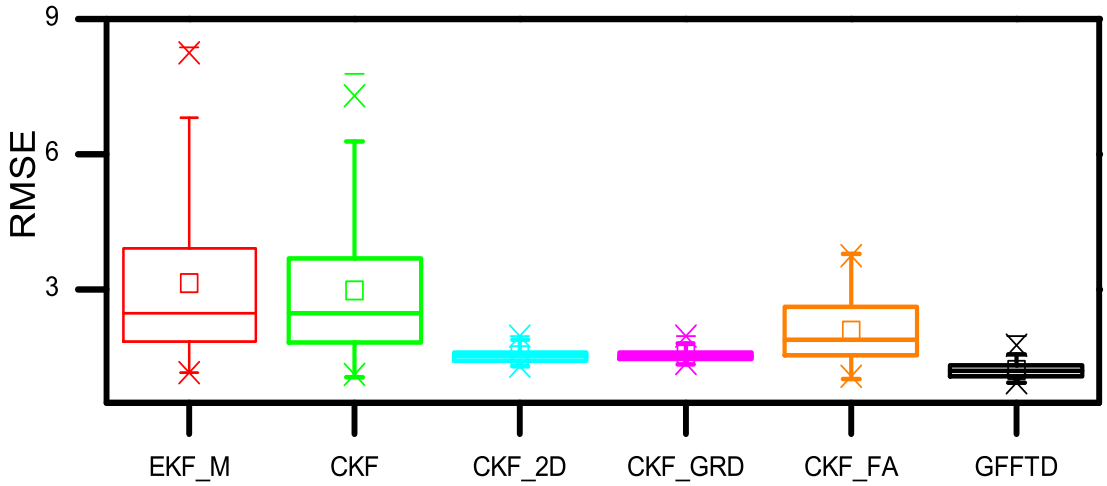
In this problem, the state space model is considered similar to chapter 3. Here, we assign $a_m = 0.5$, $b_m = 2.5$, $c_m = 2$, $\tau = 1$, while $\mathbf{Q}_k = 2$ and $\mathbf{R}_k = 0.5$. The initial true and estimate are taken as $\mathbf{x}_{0|0} = 0.2$ and $\hat{\mathbf{x}}_0 = 2\mathcal{N}(\mathbf{x}_{0|0}, P_{0|0})$, respectively, with the initial covariance $P_{0|0} = 1$. The simulation is performed for $T_S = 100$ time steps and 500 Monte-Carlo runs.

Fig. 5.2(a) shows the box plots of RMSEs obtained for all filters, while we compare the average RMSEs (ARMSE) of all filters under three different cases in Fig. 5.3(a). Please note that the figures do not include the results for the MLCKF, as it failed to execute. The figures infer a reduced RMSE of the GFFTD for all cases, which concludes the improved accuracy of the GFFTD under different system environments. The relative computational times of CKF, EKF_M, CKF_2D, CKF_FA, CKF_GRD, and GFFTD are observed as 1, 0.9932, 1.0280, 1.0031, 1.0288, and 1.0291, respectively, which conclude that the computational time of the GFFTD remains similar to the existing filters.

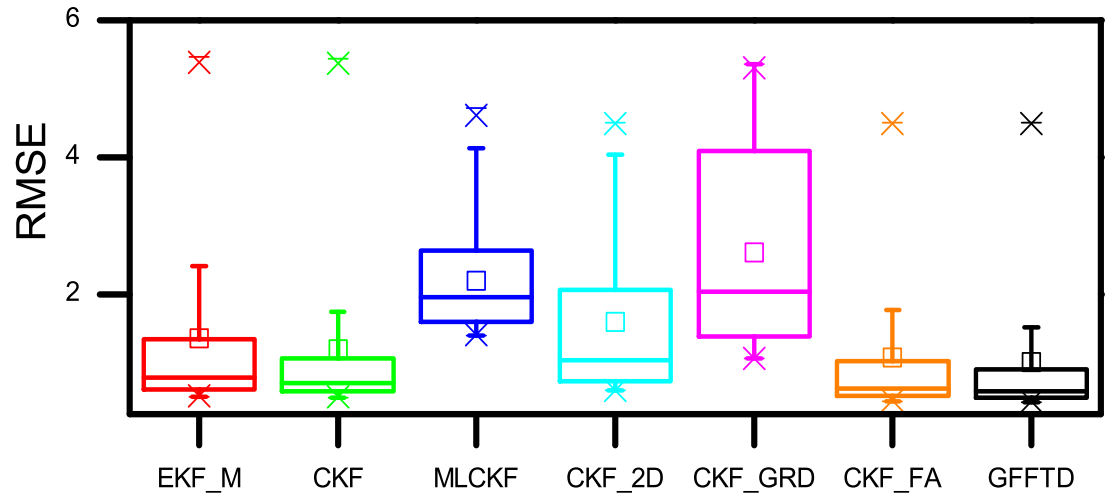
Problem 2

This chapter also looks at a second simulation problem discussed in Chapter 3, whereas $\tau = 0.25\ ms$ is taken as the sampling time. We assign $\mathbf{Q}_k = diag([\sigma_f^2\ \sigma_f^2\ \sigma_f^2\ \sigma_a^2\ \sigma_a^2\ \sigma_a^2])$ and $\mathbf{R}_k = diag([\sigma_r^2\ \ \sigma_r^2])$, with $\sigma_f = \sqrt{25}\ mHz$, $\sigma_a = \sqrt{0.8}\ mV$, and $\sigma_r = \sqrt{0.9}\ V$. Moreover, we consider $\mathbf{x}_0 = [200, 500, 1000, 3, 4, 3]^T$ and $\hat{\mathbf{x}}_{0|0} = \mathcal{N}(\mathbf{x}_0, \mathbf{P}_{0|0})$, while $\mathbf{P}_{0|0} = diag\big([20, 20, 20, 0.5, 0.5, 0.5]\big)$. The filters are implemented for $T_S = 400$ time steps and 500 Monte-Carlo runs. In this problem, we compute the RMSE as the mean of the RMSEs of the three sinusoids.

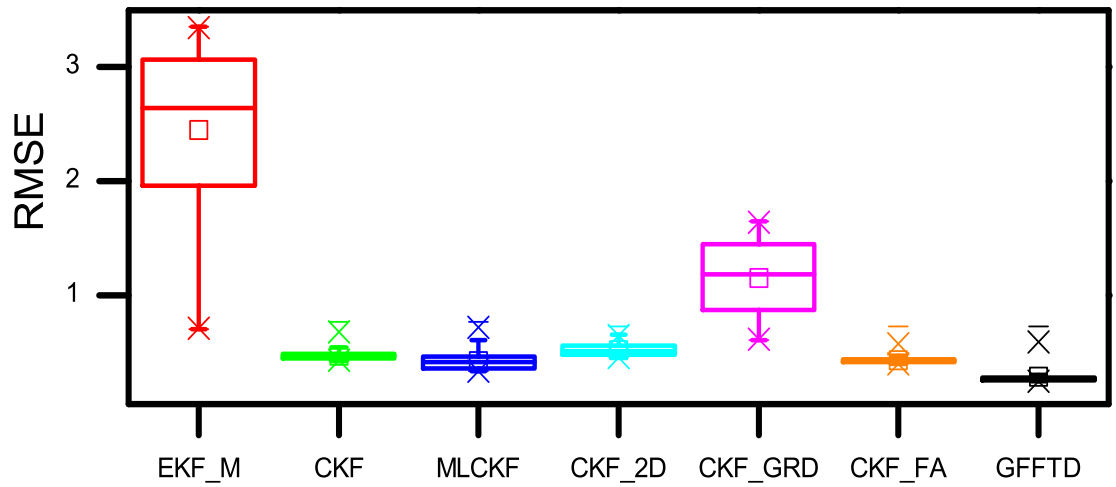
We show the box plots of the RMSEs of all filters in Figs. 5.2(b) and 5.2(c), while Fig. (5.3(b)) compares the ARMSEs of frequency for all filters under three different cases. The ARMSEs of amplitude follow a similar pattern as the frequency, being the reason to exclude its plots. Similar to Problem 1, the figures conclude the improved accuracy of the GFFTD in comparison to all existing filters. The relative computation times of CKF, EKF_M, CKF_2D, MLCKF, CKF_FA, CKF_GD, and GFFTD are 1, 0.9847, 1.3254, 1.4937, 1.3253, 1.3278, and 1.3289, respectively. This infers that the computational time of the GFFTD is marginally increased in comparison to the traditional CKF, while the computational time remains similar to the existing extensions of the CKF for handling different forms of cyber-attacks.



(a)

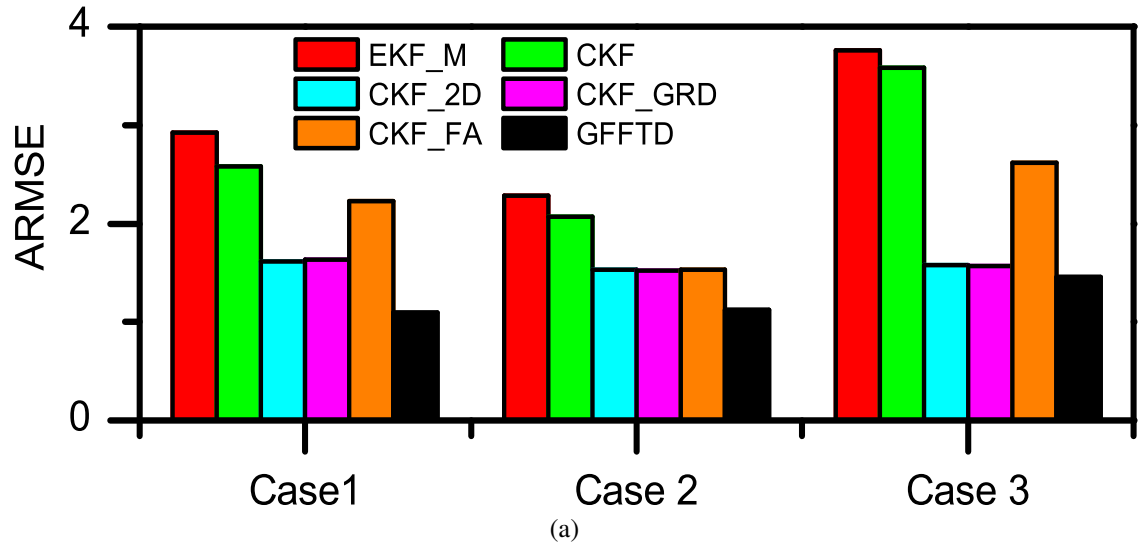


(b)

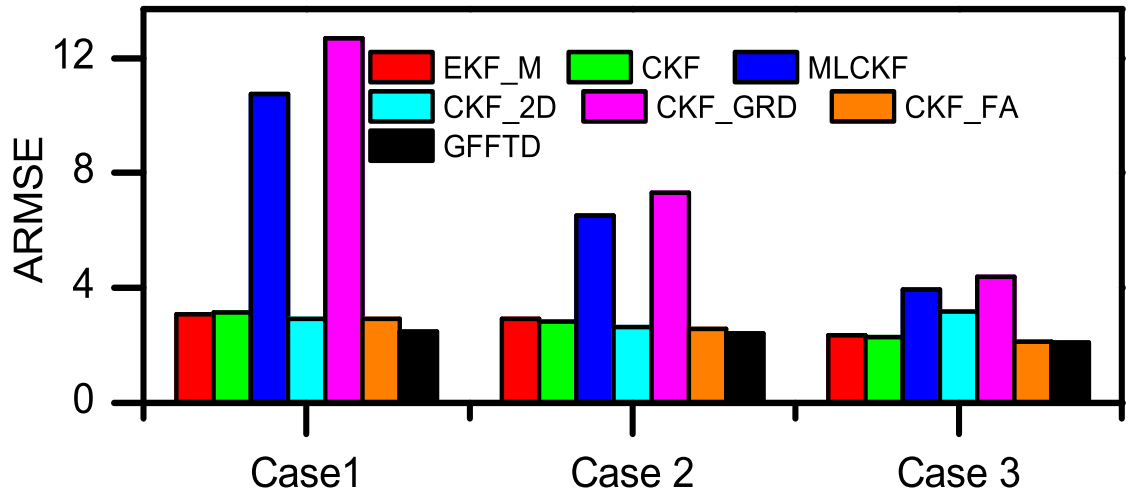


(c)

Figure 5.2: Box plots of RMSEs with $p_n = p_f = p_d = 0.333$ and $p_g = 0.5$: a) Problem 1, b) Problem 2: frequency, and c) Problem 2: amplitude.



(a)



(b)

Figure 5.3: ARMSE comparison all filters under three different cases: a) Problem 1, b) Problem 2: frequency.

We further extend the comparative analysis for other popular and more accurate Gaussian filters, including the CQKF and GHF. In this regard, in Table 5.2, we provide the ARMSEs obtained for different filters for Case 1, other cases results resemble the expected outcome, where GFFTD (CQKF) and GFFTD (GHF) denote the CQKF and GHF-based extensions of the GFFTD, respectively. Note that CQKF and GHF are implemented using the second-order quadrature rule. As expected, the table shows a reduced ARMSE for the CQKF and GHF-based extensions of the GFFTD. The findings indicate that the efficacy of the GFFTD can be improved by refining numerical computation approximation accuracy. This concludes our previous statement that any of the existing Gaussian filters can be extended under the proposed method without loosing the

generality.

Table 5.2: Case 1: Performance of Gaussian filters.

Filters	Problem 1	Problem 2	
		Frequency	Amplitude
EKF_M	2.4221	3.0987	0.8376
CKF	2.1880	3.1449	1.0110
MLCKF	NA	10.7395	1.7170
CKF_2D	1.6217	2.9257	0.8714
CKF_GD	1.6478	12.6911	3.2460
CKF_FA	1.9759	2.9286	0.9563
GFFTD (CKF)	0.9988	2.4759	0.7118
GFFTD (CQKF)	0.9799	2.0920	0.6011
GFFTD (GHF)	0.9836	2.0919	0.6010

NA: Not Available

Performance analysis for Dos and TAM attacks only

As discussed previously, the DoS and TAM attacks cause delay and missing measurements, respectively. Such measurement irregularities are caused not only due to cyber-attacks but they may be caused due to other system and data processing faults as well. For example, as discussed in chapter 2, the network faults may cause delay and missing measurements.

Interestingly, with some simplification, the GFFTD can be reformulated for handling DoS and TAM attacks only, ignoring the FDI attacks. Alternatively, with some simplification, the GFFTD can be reformulated to handle the delayed and missing measurements simultaneously. Let us name this simplified filter as GFDM in its abbreviated form, representing Gaussian filtering with delayed and missing measurements. The GFDM is also a significant development, as the existing methods for handling the delayed and missing measurements can handle only one of these irregularities and they fail for the simultaneous occurrences of the two irregularities. From Table 5.1, we can infer that considering $\beta_k = 1$ ensures the non-existence of the FDI attack. Thus, simplifying all expressions of the GFFTD by conveniently substituting $\beta_k = 1$ and $p_f = 1$, wherever required, we obtain GFDM.

We assess the performance of the GFDM for its CKF-based formulation. The assessment is performed for Problem 2, ignoring the single dimensional Problem 1. As the benchmark filters, in addition to the above used ordinary CKF, CKF_GRD, and MLCKF, we use [97] and [110], which are particularly designed for handling the delayed and missing measurements. We abbreviate the

CKF-based formulations of [97] and [110] as CKF_RD and MDCKF, respectively.

For the performance assessment, we define two new scenarios as i) Case 1: $p_n = 0.2$ and $p_d = 0.8$ and ii) Case 2: $p_n = 0.5$ and $p_d = 0.9$. Subsequently, we compare the ARMSEs for all filters, choosing p_g as 0.1, 0.5, and 0.8 for both scenarios. As can be seen in Table 5.3, the ARMSEs increase as the large delay occurrences are more prevalent (the lower value of p_g suggests more frequent large delays). Alternatively, the filtering accuracy of all filters deteriorate if the measurements are prone to large delay extents (which is an anticipated behavior). Apparently, the ARMSEs for the GFDM remain significantly smaller than those of other filters. The reduced ARMSE validates the improved accuracy of the GFDM in the presence of delayed and missing measurements.

Table 5.3: ARMSE comparison against different values of p_g (which dictates the recurrence of large delay occurrence).

States	Filters	Case 1			Case 2		
		0.1	0.5	0.8	0.1	0.5	0.8
Amplitude	CKF	6.9180	4.2454	3.0408	4.6728	3.1245	2.2408
	CKF_RD	6.0307	3.3372	2.1806	2.5382	1.2919	0.9958
	CKF_GRD	1.9155	1.3569	1.2040	1.3633	1.1247	1.0047
	MDCKF	5.8999	2.6472	1.3424	2.1868	1.1286	1.2681
	MLCKF	5.0371	2.2541	1.9775	4.1684	1.9197	1.6509
	GFDM	1.8859	1.2346	0.9836	1.3161	1.0941	0.9286
Frequency	CKF	27.1237	18.6639	15.1172	11.5302	11.6929	10.5876
	CKF_RD	21.5206	14.5804	11.4044	8.2715	5.0834	4.5953
	CKF_GRD	11.4831	6.3696	4.8383	6.1856	4.9796	4.3248
	MDCKF	25.2176	15.2685	11.7082	8.3274	5.1715	4.5890
	MLCKF	33.6225	10.8192	8.8122	24.8208	13.8789	8.2256
	GFDM	10.4981	5.8550	4.6009	6.1780	4.9747	4.3197

5.5 Summary

In this chapter, a new Gaussian filtering method, abbreviated as GFFTD, is inferred for handling simultaneously occurring FDI, TAM, and DoS forms of cyber-attack. The GFFTD re-derives the traditional Gaussian filtering method for a new formulation of the measurement model, incorporating the different forms of cyber-attack. The GFFTD is probably the first nonlinear filter to handle the simultaneously occurring FDI, TAM, and DoS attacks. The GFFTD is applicable for any of the existing Gaussian filters, such as the UKF, CKF, and GHF. The improved accuracy of the GFFTD is validated for two simulation problems.

Chapter 6

Wrapped Particle Filtering for Angular Data

6.1 Introduction

As discussed through Chapters 1 and 2, the particle filtering is a more accurate but computationally inefficient alternative of Gaussian filtering. It characterizes the desired posterior PDFs with a weighted summation of stochastically generated particles. The particles are generated from proposal density, characterizing the posterior PDFs. As the posterior PDFs are unknown, the proposal density is often taken as Gaussian. Thereby, in order to reduce the degeneracy phenomena, re-sampling techniques are adopted. The Gaussian-approximated proposal density is characterized using the first two moments.

The proposal density may be defined locally or globally [78], [160][161]. A globally defined proposal density generates all the particles at each time step [160]. Meanwhile, a locally defined proposal density is used for generating a single particle and we require a bank of local proposal densities for generating all the particles. Some of the popular particle filtering techniques, classified in terms of the underlying proposal density used, include the particle filter with transition density as its proposal (PF) [75], unscented particle filter (UPF) [78], cubature particle filter (CPF) [79], and Gauss-Hermite particle filter (GHPF) [80].

The particle filtering accuracy depends on the appropriateness of the proposal density and the re-sampling techniques. The choice of proposal density is further affected by: i) whether its

shape is close to the unknown posterior PDF and ii) whether any multivariate integrals involved in computing its moments can be computed accurately in real time. The literature primarily uses Gaussian shape for the proposal density, while many contributions appeared in the literature by introducing different numerical approximation techniques [78, 79, 80, 77, 162, 161, 163]. Similarly, the literature also witnesses many developments by advancing the resampling techniques [164, 165]. As we will observe in the later parts of this chapter, our objective is to improve the shape of the proposal density to better match the anticipated shape of the unknown PDFs. This will also require to modify the numerical approximation technique for computing the moments involved.

In all the above cases, Gaussian proposal density is over the entire real line as its support. Conversely, Gaussian distribution fails to provide a close approximation of the proposal density, if the variables themselves are constrained to a smaller support, *e.g.* $(-\pi, \pi]$ [166]. Thus we need a proposal density specifically designed for angular data.

As discussed in chapter 2, the literature behold some preliminary developments [147], [140], [141], [142]. An early development [140] utilized a truncated Fourier series with wrapped normal (WN) distribution. Nonetheless, finite-length truncation of the Fourier series affects estimation accuracy. Some of the later developments [141], [142] are designed only for linear system models in angular data. In a further development, Kurz et al. introduced a nonlinear circular filtering method through a series of publications [144], [145], [147]. Moreover, this method is designed for univariate systems, whereas many of the real-life filtering problems are multivariate. Considering the several limitations of the existing methods, efficient angular filtering is still challenging.

In this chapter, we propose a novel particle filtering method for handling nonlinear multivariate angular filtering problems. We represent the unknown angular proposal density appearing over $(-\pi, \pi]$ with WN distribution, which is a counterpart of the ordinary normal distribution in the range $(-\pi, \pi]$ [167]. The parameters of the WN distribution are the mean and the covariance. In the proposed method, the computation of the mean and the covariance involves intractable integrals of the form '*nonlinear function* $\times$ *wrapped normal distribution*'. We use univariate Rogers-Szegő quadrature rule [168], [169] for approximating such intractable integrals and also extend it to the multivariate case using the product rule. Subsequently, we name the proposed filtering method as Rogers-Szegő particle filter (RSPF). We develop and test the RSPF for both the locally and globally generated proposal densities. Further, we simulate the RSPF for two

angular filtering problems and validate its improved accuracy relative to the existing Gaussian proposal density filters from the simulation results.

Summarizing the above discussion, we highlight the following contributions of this chapter in comparison to the existing literature on filtering.

- The RSPF considers wrapped normally distributed proposal density, instead of the Gaussian proposal density which is most commonly used in the existing particle filtering algorithms.
- The RSPF utilizes Rogers-Szegő quadrature rule for the numerical approximation of the integrals. To the author's knowledge, this is the first use of this quadrature rule in particle filtering context.
- We demonstrate through comprehensive numerical examples that the RSPF can accurately handle angular data, where the existing filters underperform.
- We formulated the RSPF for both the locally and globally generated proposal density. To the author's knowledge, this is the first explicit comparison of the two different particle filtering paradigms in the literature, which offer a compromise between estimation accuracy and computational cost for the same chosen proposal density.

6.2 Problem Formulation

Our objective is to develop a particle filtering method that is suitable for angular data. The underlying state space model is represented as

$$\boldsymbol{\theta}_k = \phi_{k-1}(\boldsymbol{\theta}_{k-1}) + \nu_{k-1}, \tag{6.1}$$

$$\mathbf{y}_k = \Psi_k(\boldsymbol{\theta}_k) + \eta_k, \tag{6.2}$$

where $\boldsymbol{\theta}_k \in \mathbb{D}^n$ and $\mathbf{y}_k \in \mathbb{D}^r$, with $D \in (-\pi, \pi]$, are state and measurement variables, respectively at k^{th} instant with $k \in \{1, 2, \cdots\}$. $\phi_k : \mathbb{D}^n \to \mathbb{D}^n$ and $\Psi_k : \mathbb{D}^n \to \mathbb{D}^r$ are general nonlinear functions, representing the state dynamics and the measurement equation, respectively. Finally, $\nu_k \in \mathbb{D}^n$ and $\eta_k \in \mathbb{D}^r$ represent the process and measurement noises, respectively. Hence, the objective is to recursively estimate $\boldsymbol{\theta}_k$ $\forall k \in \{1, 2, \cdots\}$ from a sequentially received set of measurements $\mathbf{y}_k$ $\forall k \in \{1, 2, \cdots\}$.

The estimation of $\boldsymbol{\theta}_k$ from $\mathbf{y}_k$ requires characterizing the PDF $P(\boldsymbol{\theta}_k|\mathbf{y}_{1:k})$ analytically. In this regard, the particle filter [75] approximates $P(\boldsymbol{\theta}_k|\mathbf{y}_{1:k})$ as a weighted summation of stochastically generated sample points, also known as particles. Particles are sampled from an appropriately chosen proposal density $q(\boldsymbol{\theta}_k|\mathbf{y}_{1:k})$ for representing the unknown PDF $P(\boldsymbol{\theta}_k|\mathbf{y}_{1:k})$. We denote $\boldsymbol{\theta}_k^i$ and ω_k^i, $\forall i \in \{1,2,\cdots,N_p\}$, as the i^{th} particle and the associated weight, respectively, at k^{th} instant. The weights are normalized, *i.e.*, $\sum_{i=1}^{N_p} \omega_k^i = 1$. Subsequently, the desired PDF $P(\boldsymbol{\theta}_k|\mathbf{y}_{1:k})$ is approximated as

$$P(\boldsymbol{\theta}_k|\mathbf{y}_{1:k}) \approx \sum_{i=1}^{N_p} \omega_k^i \delta\left(\boldsymbol{\theta}_k - \boldsymbol{\theta}_k^i\right), \tag{6.3}$$

where $P(\mathbf{y}_k|\boldsymbol{\theta}_k^i)$ represents the likelihood function, $\delta(\cdot)$ represents dirac delta function, and N_p represents the number of particles. If $q(\boldsymbol{\theta}_k|\mathbf{y}_{1:k})$ is the same as the transition density $P(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1})$, then the weights can be recursively updated as $\omega_k^i \propto \omega_{k-1}^i P(\mathbf{y}_k/\boldsymbol{\theta}_k^i)$.

To this end, using transition density as a proposal ignores the information inferred from the latest measurement, and can result in a poor approximation of $P(\boldsymbol{\theta}_k|\mathbf{y}_{1:k})$ [79]. Using the proposal density resulting from approximate Gaussian filters such as the UKF [78] or the CKF [79], we can include the latest measurements information during the generation of particles. There are two fundamentally different approaches for generating particles. In *local sampling*, [78] we approximate $q(\boldsymbol{\theta}_k^i|\boldsymbol{\theta}_{0:k-1}^i,\mathbf{y}_{1:k}) = P(\boldsymbol{\theta}_k^i|\boldsymbol{\theta}_{k-1}^i,\mathbf{y}_{1:k}) \approx \mathcal{N}(\hat{\boldsymbol{\theta}}_{k|k}^i,\mathbf{P}_{k|k}^i)$ for each particle, where $\mathcal{N}(\cdot)$ denotes the Gaussian distribution and $\hat{\boldsymbol{\theta}}_{k|k}^i$, $\mathbf{P}_{k|k}^i$, respectively, represent i^{th} posterior mean and the posterior covariance estimates at time k. This update of density for each individual particle effectively involves running a bank of N_p individual Gaussian filters. Alternatively, [160], we can approximate $q(\boldsymbol{\theta}_k^i|\boldsymbol{\theta}_{0:k-1}^i,\mathbf{y}_{1:k}) = P(\boldsymbol{\theta}_k^i|\boldsymbol{\theta}_{k-1}^i,\mathbf{y}_{1:k}) \approx \mathcal{N}(\hat{\boldsymbol{\theta}}_{k|k},\mathbf{P}_{k|k})$ for all the particles. This requires a single Gaussian proposal at each time step and is termed as *global sampling*. The author has not come across an explicit comparison of global and local approaches to proposal density generation. Most of the subsequent discussion in this chapter follows the local sampling strategy. Therefore, we also compare our results using the global sampling strategy with those using the local sampling strategy in simulation examples.

Note that Gaussian proposal density characterizes any data over the entire real line. This is an inappropriate characterization of the angular data bounded within $(-\pi,\pi]$ and can result in poor accuracy for angular filtering. Here, we propose a novel particle filtering algorithm for angular filtering by characterizing the angular PDFs over $(-\pi,\ \pi]$ instead of the entire real line. As discussed above, we also illustrate the developed particle filtering method for both the locally

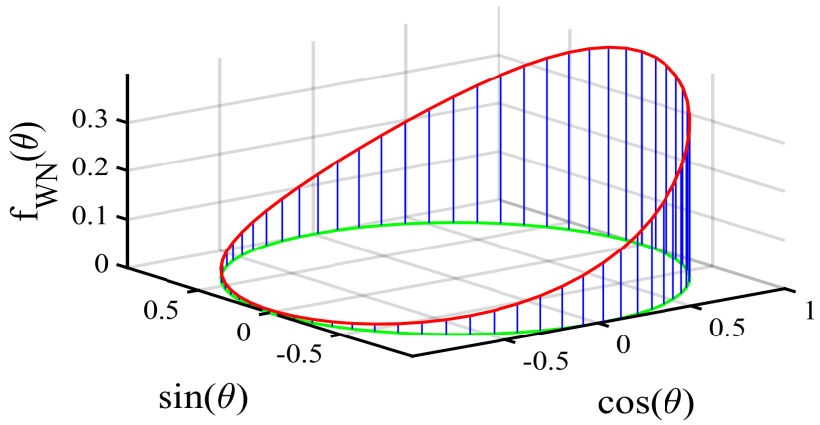


Figure 6.1: Wrapped normal distribution plot for zero-mean and unity-variance.

and globally defined proposal densities.

6.3 Rogers-Szegő Particle Filter

In this section, we introduce the RSPF, which has the potential to be far more accurate for angular data appearing over $(-\pi, \pi]$. We approximate the unknown proposal density with wrapped normal distribution [169], defined over $(-\pi, \pi]$. Before proceeding further, we briefly discuss this distribution in the next sub-section.

6.3.1 Wrapped Normal Distribution

The wrapped normal distribution may be obtained by wrapping the horizontal axis of an ordinary normal distribution curve around a unit circle [147]. Similar to a normal distribution, a wrapped normal distribution is also completely characterized by the mean and the variance. If θ is wrapped normally distributed with mean μ and variance σ^2, then the distribution of θ is given as [144],

$$f_{WN}(\theta;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \sum_{l=-\infty}^{\infty} \exp\left(\frac{-(\theta-\mu-2\pi l)^2}{2\sigma^2}\right). \tag{6.4}$$

The zero-mean and unit-variance wrapped normal distribution is plotted in Fig. 6.1. We refer to [170, 144, 143] for a more detailed discussion on the wrapped normal distribution.

6.3.2 Proposal Density Approximation

As stated in [170, 144], the wrapped normal distribution follows the circular central limit theorem for angular data. Thus, we consider the following approximations.

- The prior and posterior distributions of $\boldsymbol{\theta}_k^i$ are approximated as wrapped normally distributed, *i.e.*, $P(\boldsymbol{\theta}_k^i|\mathbf{y}_{k-1}) \approx \mathbf{f_{WN}}(\boldsymbol{\theta}_k^i; \hat{\boldsymbol{\theta}}_{k|k-1}^i, \mathbf{P}_{k|k-1}^i)$ and $P(\boldsymbol{\theta}_k^i|\mathbf{y}_k) \approx \mathbf{f_{WN}}(\boldsymbol{\theta}_k^i; \hat{\boldsymbol{\theta}}_{k|k}^i, \mathbf{P}_{k|k}^i)$, where $\mathbf{f_{WN}}(\cdot)$ represents multivariate wrapped normal distribution. Here, $\hat{\boldsymbol{\theta}}_{k|k}^i$, $\mathbf{P}_{k|k}^i$, respectively, represent the i^{th} posterior mean and the posterior covariance of the state at time k.

- We further approximate $P(\mathbf{y}_k^i|\mathbf{y}_{k-1}) \approx \mathbf{f_{WN}}(\mathbf{y}_k^i; \hat{\mathbf{y}}_{k|k-1}^i, \mathbf{P}_{k|k-1}^{\mathbf{yy}i})$, where $\hat{\mathbf{y}}_{k|k-1}^i$ and $\mathbf{P}_{k|k-1}^{\mathbf{yy}i}$ denote the mean and the covariance of $\mathbf{y}_k^i$, respectively.

- The desired proposal density is approximated as wrapped normally distributed, i.e., $q(\boldsymbol{\theta}_k^i|\boldsymbol{\theta}_{0:k-1}^i, \mathbf{y}_{1:k}) \approx \mathbf{f_{WN}}(\boldsymbol{\theta}_k^i; \hat{\boldsymbol{\theta}}_{k|k}^i, \mathbf{P}_{k|k}^i)$.

- The noises, η_k and ϑ_k, are approximated as $\mathbf{f_{WN}}(\mathbf{0}, \mathbf{Q}_k)$ and $\mathbf{f_{WN}}(\mathbf{0}, \mathbf{R}_k)$, respectively, where $\mathbf{Q}_k$ and $\mathbf{R}_k$ are the covariances of the respective noises.

- Additionally, we assume that the noises, η_k and ϑ_k are independent of each other as well as serially independent.

From the above assumptions, the proposal density can be characterized by mean $\hat{\boldsymbol{\theta}}_{k|k}^i$ and covariance $\mathbf{P}_{k|k}^i$ for each particle, which corresponds to local sampling in our earlier discussion. As in any standard Gaussian filtering algorithm, we obtain $\hat{\boldsymbol{\theta}}_{k|k}^i$ and $\mathbf{P}_{k|k}^i$ in two steps, prediction and update. The prediction characterizes the PDF $\mathbf{P}(\boldsymbol{\theta}_k^i|\mathbf{y}_{k-1})$ by computing the mean $\hat{\boldsymbol{\theta}}_{k|k-1}^i$ and covariance $\mathbf{P}_{k|k-1}^i$. The update characterizes the desired PDF $P(\boldsymbol{\theta}_k^i|\mathbf{y}_k)$ by determining the mean $\hat{\boldsymbol{\theta}}_{k|k}^i$ and the covariance $\mathbf{P}_{k|k}^i$ approximately, using the formulae for conditional mean and the conditional variance for Gaussian distributions, which is a step similar to other assumed Gaussian density filters [22, 143]. We provide the computational aspects of these two steps in the following discussion.

Prediction

This step computes $\hat{\boldsymbol{\theta}}^i_{k|k-1}$ and $\mathbf{P}^i_{k|k-1}$, respectively, as [80]

$$\hat{\boldsymbol{\theta}}^i_{k|k-1}=\int_{\mathbb{D}^n} \phi_{k-1}(\boldsymbol{\theta}^i_{k-1})\mathbf{f_{WN}}(\boldsymbol{\theta}^i_{k-1};\hat{\boldsymbol{\theta}}^i_{k-1|k-1},\mathbf{P}^i_{k-1|k-1})d\boldsymbol{\theta}^i_{k-1}, \tag{6.5}$$

$$\begin{aligned}\mathbf{P}^i_{k|k-1} = &\int_{\mathbb{D}^n} \phi_{k-1}(\boldsymbol{\theta}^i_{k-1})\phi^T_{k-1}(\boldsymbol{\theta}^i_{k-1})\mathbf{f_{WN}}(\boldsymbol{\theta}^i_{k-1};\hat{\boldsymbol{\theta}}^i_{k-1|k-1},\mathbf{P}^i_{k-1|k-1})d\boldsymbol{\theta}^i_{k-1} \\ &- \hat{\boldsymbol{\theta}}^i_{k-1|k-1}\hat{\boldsymbol{\theta}}^{iT}_{k-1|k-1} + \mathbf{Q}^i_k.\end{aligned} \tag{6.6}$$

Update

This step computes $\hat{\boldsymbol{\theta}}^i_{k|k}$ and $\mathbf{P}^i_{k|k}$, respectively, as [80]

$$\hat{\boldsymbol{\theta}}^i_{k|k} = \hat{\boldsymbol{\theta}}^i_{k|k-1} + \mathbf{K}^i_k(\mathbf{y}_k - \hat{\mathbf{y}}^i_{k|k-1}), \tag{6.7}$$

$$\mathbf{P}^i_{k|k} = \mathbf{P}^i_{k|k-1} - \mathbf{K}^i_k P^{\mathbf{yyi}}_{k|k-1} K^{iT}_k, \tag{6.8}$$

where $\hat{\mathbf{y}}^i_{k|k-1}$, $\mathbf{P}^{\mathbf{yy}i}_{k|k-1}$, $\mathbf{P}^{\boldsymbol{\theta}\mathbf{y}i}_{k|k-1}$, and $\mathbf{K}^i_k$ are determined, respectively, as [80]

$$\hat{\mathbf{y}}^i_{k|k-1} = \int_{\mathbb{D}^n} \Psi_k(\boldsymbol{\theta}^i_k)\mathbf{f_{WN}}(\boldsymbol{\theta}^i_k;\hat{\boldsymbol{\theta}}^i_{k|k-1},\mathbf{P}^i_{k|k-1})d\boldsymbol{\theta}^i_k, \tag{6.9}$$

$$\mathbf{P}^{\mathbf{yy}i}_{k|k-1} = \int_{\mathbb{D}^n} \Psi_k(\boldsymbol{\theta}^i_k)\Psi^T_k(\boldsymbol{\theta}^i_k)\mathbf{f_{WN}}(\boldsymbol{\theta}^i_k;\hat{\boldsymbol{\theta}}^i_{k|k-1},\mathbf{P}^i_{k|k-1})d\boldsymbol{\theta}^i_k - \hat{\mathbf{y}}^i_{k|k-1}\hat{\mathbf{y}}^{iT}_{k|k-1} + \mathbf{R}^i_k, \tag{6.10}$$

$$\mathbf{P}^{\boldsymbol{\theta}\mathbf{y}i}_{k|k-1} = \int_{\mathbb{D}^n} \boldsymbol{\theta}^i_k\Psi^T_k(\boldsymbol{\theta}^i_k)\mathbf{f_{WN}}(\boldsymbol{\theta}^i_k;\hat{\boldsymbol{\theta}}^i_{k|k-1},\mathbf{P}^i_{k|k-1})d\boldsymbol{\theta}^i_k - \hat{\boldsymbol{\theta}}^i_{k|k-1}\hat{\mathbf{y}}^{iT}_{k|k-1}, \tag{6.11}$$

$$\mathbf{K}^i_k = \mathbf{P}^{\boldsymbol{\theta}\mathbf{y}i}_{k|k-1}(\mathbf{P}^{\mathbf{yy}i}_{k|k-1})^{-1}. \tag{6.12}$$

Eqs. (6.5) to (6.12) give the steps needed to generate the i^{th} mean-covariance pair $\left(\hat{\boldsymbol{\theta}}^i_{k|k}, P^i_{k|k}\right)$, given $\left(\hat{\boldsymbol{\theta}}^i_{k-1|k-1}, P^i_{k-1|k-1}\right)$ and the measurement $\mathbf{y}_k$. Note that these equations use the same approximate equations for prior and posterior distributions as in the case of other assumed Gaussian density filters, apart from one crucial difference that wrapped normal density is used in place of normal density for calculating moments.

The characterization of the proposal density using the above equations requires computing

integrals of the form

$$I^n(\mathbb{F}) = \int_{\mathbb{D}^n} \mathbb{F}(\boldsymbol{\theta}) \mathbf{f}_{\mathbf{WN}}(\boldsymbol{\theta}; \hat{\boldsymbol{\theta}}, \mathbf{P}) d\boldsymbol{\theta}, \tag{6.13}$$

where $\mathbb{F} : \mathbb{D}^n \to \mathbb{D}^n$ is a general nonlinear function. As a closed-form solution is not available for such integrals in general, the RSPF introduces Rogers-Szegő quadrature rule [169] for approximating such integrals. The integrals occurring here are different from those which occur in traditional proposal densities or in approximate Gaussian filters [22] or in traditional proposal densities for particle filters [79, 78, 80], since these involve a Gaussian density (rather than a wrapped normal density).

6.3.3 Approximation of Integrals

In this section, we introduce the Rogers-Szegő quadrature rule for approximating the desired intractable integral $I^n(\mathbb{F})$. While the Rogers-Szegő quadrature rule is applicable only for univariate systems, the desired integral $I^n(\mathbb{F})$ is multivariate. We additionally use the product rule [60] for extending the univariate Rogers-Szegő quadrature rule to the multivariate case.

We will first approximate the standard form (zero-mean and unity-covariance) of $I^n(\mathbb{F})$, given as

$$I_0^n(\mathbb{F}) = \int_{\mathbb{D}^n} \mathbb{F}(\boldsymbol{\theta}) \mathbf{f}_{\mathbf{WN}}(\boldsymbol{\theta}; \mathbf{0}, \mathbf{I}_n) d\boldsymbol{\theta}, \tag{6.14}$$

where $\mathbf{I}_n$ represents the identity matrix. Further, the multivariate integral $I_0^n(\mathbb{F})$ can be expressed as

$$I_0^n(\mathbb{F}) = I_0^1(\mathbb{F}_1) \times I_0^1(\mathbb{F}_2) \times \cdots \times I_0^1(\mathbb{F}_n), \tag{6.15}$$

where the univariate integrals $I_0^1(\mathbb{F}_q)\ \forall q \in \{1, 2, \cdots, n\}$ can be expressed as

$$I_0^1(\mathbb{F}_q) = \int_{-\pi}^{\pi} \mathbb{F}(\theta_q) f_{WN}(\theta_q; 0, 1) d\theta_q, \tag{6.16}$$

with

$$f_{WN}(\theta_q; 0, 1) = \sqrt{\frac{1}{2\pi}} \sum_{l=-\infty}^{\infty} \exp\left(-\frac{(\theta_q - 2\pi l)^2}{2}\right). \tag{6.17}$$

The univariate Rogers-Szegő quadrature rule is designed for approximating $I_0^1(\mathbb{F}_q)$ given in Eq.

(6.16).

Univariate Rogers-Szegő quadrature rule

The Rogers-Szegő quadrature rule, designed for approximating $I_0^1(\mathbb{F}_q)$, is defined over unit circle $\mathbb{U}$ [171]. We denote the unit circle as $\mathbb{U} = \{\mathbb{Z} \in \mathbb{C} : |\mathbb{Z}| = 1\}$, where $\mathbb{Z} = e^{j\theta_q}$, while $\mathbb{C}$ represents the set of all circles. The Rogers-Szegő quadrature rule utilizes Rogers-Szegő polynomials [168], which are orthogonal with the weight function $f_{WN}(\theta_q;0,1)$ on unit circle and also normalized, *i.e.*, $\int_{-\pi}^{\pi} f_{WN}(\theta_q;0,1)d\theta_q = 1$.

Let us denote the Rogers-Szegő polynomials as $\nu_m(\mathbb{Z})$, $\forall m \in \{1,2,\cdots\}$, which are defined on unit circle $\mathbb{U}$ [168]. The orthogonality of $\nu_m(\mathbb{Z})$ is defined in terms of inner product of $\nu_m(\mathbb{Z})$ induced by $f_{WN}(\theta_q;0,1)$. The inner product must satisfy

$$< \nu_m, \nu_{\bar{m}} >= \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} \nu_m(\mathbb{Z})\overline{\nu}_m(\mathbb{Z}) f_{WN}(\theta_q;0,1)d\theta_q = \delta_{m,\bar{m}}, \tag{6.18}$$

where $\bar{\nu}_m(\mathbb{Z})$ represents the conjugate of $\nu_m(\mathbb{Z})$ and $\delta_{m,\bar{m}}$ denotes Kronecker delta.

The Rogers-Szegő polynomials satisfying Eq. (6.18) are obtained from monic sequences evaluated from iterative solutions of Rogers-Szegő forward recurrence relation [172]. Let us denote $\gamma_m(\mathbb{Z})$, $\forall m \in \{1,2,\cdots\}$, as the monic sequences of $\nu_m(\mathbb{Z})$, then the Rogers-Szegő forward recurrence relation is given as [173],

$$\begin{bmatrix} \gamma_m(\mathbb{Z}) \\ \gamma_m^*(\mathbb{Z}) \end{bmatrix} = \begin{bmatrix} \mathbb{Z} & \beta_m \\ \bar{\beta}_m\mathbb{Z} & 1 \end{bmatrix} \begin{bmatrix} \gamma_{m-1}(\mathbb{Z}) \\ \gamma_{m-1}^*(\mathbb{Z}) \end{bmatrix}, \tag{6.19}$$

where $m \geq 1$, $\gamma_m^*(\mathbb{Z})$ is reciprocal of $\gamma_m(\mathbb{Z})$, and $\beta_m = \gamma_m(0)$ $\forall m \geq 0$ is Verblunsky coefficient [169]. The reciprocal value $\gamma_m^*(\mathbb{Z})$ is obtained as $\gamma_m^*(\mathbb{Z}) = \mathbb{Z}^m\overline{\gamma_m(1/\bar{\mathbb{Z}})}$, where $\bar{\gamma}_m$ denotes the conjugate of γ_m. As mentioned in [168], the recurrence relation can be initiated with $\gamma_0(\mathbb{Z}) = \gamma_0^*(\mathbb{Z}) = 1$. Moreover, $\beta_0 = 1$ and $|\beta_m| < 1$ $\forall m \leq 1$ [169]. The readers may refer to [168], [171] for a detailed discussion on $\gamma_m(\mathbb{Z})$, β_m, and the properties of Rogers-Szegő polynomials for the family of orthogonal polynomials on $\mathbb{U}$.

The solution of the recurrence relation (6.19) corresponding to $f_{WN}(\theta_q;0,1)$ gives the desired

monic sequences in terms of s-binomial coefficient as [169]

$$\gamma_m(\mathbb{Z}) = \sum_{j=0}^{m} (-1)^{m-j} \begin{bmatrix} m \\ j \end{bmatrix}_s s^{\frac{m-j}{2}} \mathbb{Z}^j, \tag{6.20}$$

where s-binomial coefficient is

$$\begin{bmatrix} m \\ j \end{bmatrix}_s = \frac{(m)_s}{(j)_s (m-j)_s} = \frac{\prod_{k=m-j+1}^{m} (1-s^k)}{\prod_{k=1}^{j} (1-s^k)},$$

with $0 < s < 1$ and

$$(0)_s = \begin{pmatrix} m \\ 0 \end{pmatrix}_s = \begin{pmatrix} m \\ m \end{pmatrix}_s = 1.$$

After $\gamma_m(\mathbb{Z})$ is obtained from Eq. (6.20), the m^{th} Rogers-Szegő polynomial is obtained as

$$\nu_m(\mathbb{Z}) = \frac{1}{\sqrt{(m)_s}} \gamma_m(\mathbb{Z}), \tag{6.21}$$

where

$$(m)_s = \prod_{i=1}^{m} (1-s^j).$$

From [171], $s = e^{-1}$ corresponds to the desired weight function $f_{WN}(\theta_q; 0, 1)$. Utilizing the above interpretations, [169] derives and states that the N_q number of desired Rogers-Szegő quadrature points can be obtained as the phase of complex roots of the polynomial

$$A_{N_q}(\mathbb{Z}) = \sum_{j=0}^{N_q} B_j \left[1 + \delta_m (-1)^{N_q} s^{(j-N_q/2)} \right] \mathbb{Z}^j, \tag{6.22}$$

where $|\delta_m| = 1$ and

$$B_j = (-1)^{N_q - j} \begin{bmatrix} N_q \\ j \end{bmatrix}_s s^{\frac{N_q - j}{2}},$$

with $j \in \{1, 2, \cdots, N_q\}$. Let us denote the roots of $A_{N_q}(\mathbb{Z})$ as λ_j, $\forall j \in \{1, 2, \cdots, N_q\}$. Then, the j^{th} Rogers-Szegő quadrature point is obtained as $\Upsilon_j = \angle \lambda_j \ \forall j \in \{1, 2, \cdots, N_q\}$. Eventually, the weight W_j associated with Υ_j can be obtained in terms of monic sequence γ_{N_q} as [169]

$$W_j = \frac{(N_q)_s}{2Re[\lambda_j \gamma'_{N_q}(\lambda_j)\overline{\gamma_{N_q}(\lambda_j)}] - N_q|\gamma_{N_q}(\lambda_j)|^2}, \tag{6.23}$$

where $Re[\cdot]$ represents the real part of complex number and γ' denotes the first derivative of γ.

After Υ_j and W_j are obtained, the univariate integral $I_0^1(\mathbb{F}_q)$ given in Eq. (6.16) can be approximated as

$$I_0^1(\mathbb{F}_q) \approx \int_{-\pi}^{\pi} \mathbb{F}(\theta_q) f_{WN}(\theta_q) d\theta_q \approx \sum_{j=1}^{N_q} \mathbb{F}(\Upsilon_j) W_j. \tag{6.24}$$

Table-6.1 illustrates the complex values of λ, the angular points Υ, and the associated weights W for eight-points univariate Rogers-Szegő quadrature rule with $s = e^{-1}$. Please note that $s = e^{-1}$ is an essential requirement for the RSPF as mentioned above.

Table 6.1: Univariate quadrature points and associated weights for $s = e^{-1}$ and $N_q = 8$.

λ	$\Upsilon = \angle\lambda$	W
- 0.82206297 $\pm$0.56939659i	$\pm$2.5358	0.015354324
- 0.15523459 $\pm$0.98787764i	$\pm$1.7267	0.066536929
0.52747007 $\pm$0.849573610i	$\pm$1.0152	0.164552400
0.94424738 $\pm$0.329236820i	$\pm$0.3355	0.253556350

Multivariate extension of Rogers-Szegő quadrature rule

In the above discussion, we introduced the univariate Rogers-Szegő rule for approximating the univariate integral $I_0^1(\mathbb{F}_q)$. We now extend the univariate Rogers-Szegő rule for approximating the multivariate integral $I_0^n(\mathbb{F})$.

The product rule states that given the univariate Rogers-Szegő quadrature points $\{\Upsilon_1, \Upsilon_2, \cdots, \Upsilon_{N_q}\}$ and the associated weights $\{W_1, W_2, \cdots, W_{N_q}\}$, we can approximate $I_0^n(\mathbb{F})$ as

$$I_0^n(\mathbb{F}) \approx \sum_{j_1=1}^{N_q} \sum_{j_2=1}^{N_q} \cdots \sum_{j_n=1}^{N_q} \mathbb{F}([\Upsilon_{j_1}, \Upsilon_{j_2}, \cdots, \Upsilon_{j_n}]) \prod_{l=1}^{n} W_{j_l}, \tag{6.25}$$

where $j_1, j_2, \cdots, j_n \in \{1, 2, \cdots, N_q\}$.

Finally, we can approximate the desired intractable integral I^n corresponding to $\mathbf{f_{WN}}(\boldsymbol{\theta}; \hat{\boldsymbol{\theta}}, \mathbf{P})$,

as

$$I^n \approx \sum_{j_1=1}^{N_q} \sum_{j_2=1}^{N_q} \cdots \sum_{j_n=1}^{N_q} \mathbb{F}\left(\hat{\boldsymbol{\theta}} + \boldsymbol{\Sigma}[\Upsilon_{j_1}, \Upsilon_{j_2}, \cdots, \Upsilon_{j_n}]\right) \prod_{l=1}^{n} W_{j_l}. \tag{6.26}$$

We can rewrite the above expression as

$$I^n \approx \sum_{j=1}^{N_s} \mathbb{F}\left(\hat{\boldsymbol{\theta}} + \boldsymbol{\Sigma}\boldsymbol{\Upsilon}_j\right) \mathbf{W}_j, \tag{6.27}$$

where $N_s = (N_q)^n$ denotes the number of the multivariate Rogers-Szegő quadrature points, while $\boldsymbol{\Upsilon}_j = [\Upsilon_{j_1}\Upsilon_{j_2}, \cdots, \Upsilon_{j_n}]$ and $\mathbf{W}_j = \prod_{j=1}^{n} W_{j_l}$, with $j_1, j_2, \cdots, j_n \in \{1, 2, \cdots, N_q\}$, represent the multivariate quadrature points and weights, respectively. Moreover, $\boldsymbol{\Sigma}$ represents the Cholesky decomposition of $\mathbf{P}$, *i.e.*, $\mathbf{P} = \boldsymbol{\Sigma}\boldsymbol{\Sigma}^T$.

We can use the product rule-based multivariate Rogers-Szegő quadrature rule for approximating the intractable integrals that appeared through Eqs.(6.5) to (6.12). More specifically, we can construct the proposal density $q(\boldsymbol{\theta}_k^i | \boldsymbol{\theta}_{0:k-1}^i, \mathbf{y}_{1:k}) \approx \mathbf{f}_{\mathbf{WN}}(\boldsymbol{\theta}_k^i; \hat{\boldsymbol{\theta}}_{k|k}^i, \mathbf{P}_{k|k}^i)$ by determining the mean $\hat{\boldsymbol{\theta}}_{k|k}^i$ and the covariance $\mathbf{P}_{k|k}^i$ from Eqs. (6.5) and (6.12), respectively. Considering that the multivariate quadrature points and weights, *i.e.*, $\boldsymbol{\Upsilon}_j$ and W_j, $\forall j \in \{1, 2, \cdots, N_q\}$, are available, we refer to Appendix B for analytical steps used for implementing the prediction and update steps discussed in Section 6.3.2.

6.3.4 Particle Filtering with Wrapped Proposal Distribution

We now introduce the RSPF using the wrapped normal distribution for handling angular data. The RSPF uses the proposal density determined through the subsections 6.3.1-6.3.2 in particle filtering. As discussed previously, we develop two filters, one for local sampling and another for global sampling. We abbreviate the two filters as LRSPF and GRSPF, respectively. The previous discussions of this section provides the formulations for the local sampling, *i.e.* for the LRSPF. The formulation for GRSPF is similar, although simpler due to a single filter at each time step.

LRSPF

The implementation of the LRSPF comprises the following steps.

- Initialization: We initialize the filter at $k = 0$ with initial state $\boldsymbol{\theta}_0$ and initial covariance $\mathbf{P}_{0|0}$. Subsequently, we generate the initial set of particles as $\{\boldsymbol{\theta}_0\}_{i=1}^{N_p} \sim P(\boldsymbol{\theta}_0)$ and the associated weights as $\{\omega_0^i\}_{i=1}^{N_p} = 1/N_p$.

- Particles and weights calculation at k^{th} instant ($k \geq 1$):

  Construct the initialization for k^{th} instant from the latest particles and covariance (assign $\hat{\boldsymbol{\theta}}_{k-1|k-1}^i$ and $\mathbf{P}_{k-1|k-1}^i$), and follow the steps below.

  - Propagate $\hat{\boldsymbol{\theta}}_{k-1|k-1}^i$ and $\mathbf{P}_{k-1|k-1}^i$ through Eqs. (6.5) to (6.12) to obtain $\hat{\boldsymbol{\theta}}_{k|k}^i$ and $\mathbf{P}_{k|k}^i$ for each particle. Then, the particles are updated by sampling from wrapped normal distribution as $\tilde{\boldsymbol{\theta}}_{k|k}^i \sim \mathbf{f_{WN}}(\boldsymbol{\theta}_k^i; \hat{\boldsymbol{\theta}}_{k|k}^i, \mathbf{P}_{k|k}^i)$.

  - Subsequently, the weights are updated as

$$\omega_k^i \propto \omega_{k-1}^i \frac{P(\mathbf{y}_k|\tilde{\boldsymbol{\theta}}_{k|k}^i) P(\tilde{\boldsymbol{\theta}}_{k|k}^i|\boldsymbol{\theta}_{k-1}^i)}{q(\tilde{\boldsymbol{\theta}}_k^i|\boldsymbol{\theta}_{0:k-1}^i, \mathbf{y}_{1:k})}. \tag{6.28}$$

- Weight normalization: The weights are normalized as

$$\omega_k^i = \frac{\omega_k^i}{\sum_{i=1}^N \omega_k^i}, \tag{6.29}$$

  and determine the effective sample size N_{eff} as

$$N_{eff} \approx \frac{1}{\sum_{i=1}^N (\omega_k^i)^2}. \tag{6.30}$$

- Re-sampling: If N_{eff} is below a preassigned threshold value N_{th}, then we perform re-sampling [79] and a new set of $\boldsymbol{\theta}_k^i \ \forall i \in \{1, 2, \cdots, N_p\}$ is generated from the current set of particles $\tilde{\boldsymbol{\theta}}_k^i \ \forall i \in \{1, 2, \cdots, N_p\}$. In general, we consider $N_{th} = 2/3N_p$. Furthermore, newly updated weights are obtained as, $\omega_k^i = 1/N_p$.

- Estimation: Finally, the desired estimate and covariance are obtained as

$$\tilde{\boldsymbol{\theta}}_{k|k} \approx \sum_{i=1}^{N_p} \omega_k^i \boldsymbol{\theta}_k^i, \tag{6.31}$$

$$\tilde{\mathbf{P}}_{k|k} \approx \sum_{i=1}^{N_p} \omega_k^i \left(\boldsymbol{\theta}_k^i - \tilde{\boldsymbol{\theta}}_{k|k}\right) \left(\boldsymbol{\theta}_k^i - \tilde{\boldsymbol{\theta}}_{k|k}\right)^T. \tag{6.32}$$

The LRSPF requires implementing the algorithm discussed through Eqs. 6.5 to 6.12 implemented once for each particle, which increases the computational demand.

GRSPF

In an alternate method to the LRSPF, a single proposal density is globally defined in order to reduce the computational demand. This single proposal density is used for generating all particles. The steps for implementing the GRSPF is provided below.

- Initialization: This step is the same as the LRSPF, which can be followed from the previous discussion.
- Compute the mean and covariance of particles at k^{th} instant $(k \geq 1)$:

$$\hat{\boldsymbol{\theta}}_{k-1|k-1} \approx \sum_{i=1}^{N_p} \omega_{k-1}^{i} \boldsymbol{\theta}_{k-1}^{i}, \tag{6.33}$$

$$\mathbf{P}_{k-1|k-1} \approx \sum_{j=1}^{N_p} \omega_{k-1}^{i} \left(\boldsymbol{\theta}_{k-1}^{i} - \hat{\boldsymbol{\theta}}_{k-1|k-1}\right)\left(\boldsymbol{\theta}_{k-1}^{i} - \hat{\boldsymbol{\theta}}_{k-1|k-1}\right)^{T}. \tag{6.34}$$

  - Propagate $\hat{\boldsymbol{\theta}}_{k-1|k-1}$ and $\mathbf{P}_{k-1|k-1}$ through the algorithm discussed in Section 6.3.2 to obtain $\hat{\boldsymbol{\theta}}_{k|k}$ and $\mathbf{P}_{k|k}$ for single mean and covariances.

- Compute particles: Generate the desired particles as $\{\tilde{\boldsymbol{\theta}}_{k}^{i}\}_{i=1}^{N_p} \sim \mathbf{f_{WN}}(\hat{\boldsymbol{\theta}}_{k|k}, \mathbf{P}_{k|k})$.
- Compute weights: Update the weights

$$\omega_{k}^{i} \propto \omega_{k-1}^{i} \frac{P(\mathbf{y}_k|\tilde{\boldsymbol{\theta}}_{k|k}^{i}) P(\tilde{\boldsymbol{\theta}}_{k|k}^{i}|\boldsymbol{\theta}_{k-1}^{i})}{q(\tilde{\boldsymbol{\theta}}_{k}^{i}|\boldsymbol{\theta}_{0:k-1}^{i}, \mathbf{y}_{1:k})}. \tag{6.35}$$

- The remaining steps are the same as those discussed for LRSPF.

For angular data, the RSPF with wrapped normal proposal and Rogers-Szegő quadrature rule for integration is more accurate than the numerical approximation methods used in the state-of-art particle filters such as the UPF, CPF, and GHPF under assumed Gaussian density, as amply illustrated in the next section. The LRSPF is slightly more accurate than the GRSPF due to the local sampling and may be preferred if a computational budget is available.

An appropriate distribution for angular data is probably the wrapped normal distribution, for which numerical methods can be used to approximate multivariate integrals. We have used this

intuition to develop a new particle filtering algorithm. In the next section, we demonstrate its superior performance using simulation examples on multivariate nonlinear systems where data is inherently angular.

6.4 Simulation Examples

In this section, we validate the performance of the RSPF for two angular filtering problems. We compare the performance of the LRSPF and GRSPF with various existing filters, including the particle filter with transition density as its proposal (referred to as PF) [75], unscented particle filter or the UPF [78], cubature particle filter or the CPF [79], and Gauss-Hermite particle filter or the GHPF [80], and the existing nonlinear circular filter (CF) [147]. The existing filters, UPF, CPF, and GHPF are implemented with local sampling to characterize them with their best accuracy. Note that CF is applicable only for univariate systems. Thus, it will be included in the comparison for the first simulation problem only, as the second problem is multivariate.

The LRSPF, GRSPF, and GHPF are implemented with two univariate quadrature points. The parameter $\bar{\kappa}$ for implementing the UPF is assigned as $\bar{\kappa} = 4$. Finally, CF is applied by considering a three-point wrapped Dirac mixture distribution.

The performance analysis and comparison are based on angular root mean square error (RMSE) between the true and estimated states. Note that the true and the estimated states may often fall beyond the angle range $(-\pi, \pi]$ due to noise. In such cases, we perform aliasing to obtain the equivalent angle within $(-\pi, \pi]$. The angular root mean square error (RMSE) is expressed as,

$$\text{RMSE}_k = \sqrt{\frac{1}{T_{mc}} \sum_{i=1}^{T_{mc}} \left(\min \left(|\tilde{\theta}^i_{k|k} - \theta^i_k|, 2\pi - |\tilde{\theta}^i_{k|k} - \theta^i_k| \right) \right)^2}, \tag{6.36}$$

where θ^i_k and $\tilde{\theta}^i_{k|k}$ are the true and the estimated states at k^{th} time-step and in i^{th} simulation run, and T_{mc} is the number of Monte-carlo simulations.

6.4.1 Problem 1

This is an angle estimation problem of a robot's arm [147], which has applications in robotics often require closed-loop position control of the robot arm, requiring a measurement of the current angle of the arm as feedback. This, in turn, involves an estimate of an angular quantity from noisy

sensor measurements. The dynamic state space model of the system is given as [147]

$$\theta_k = \theta_{k-1} + d_1 \sin(\theta_{k-1}) + d_2 + \eta_{k-1}, \tag{6.37}$$

$$\mathbf{y}_k = \begin{bmatrix} \sin(\theta_k) \\ \cos(\theta_k) \end{bmatrix} + \vartheta_k, \tag{6.38}$$

where d_1 and d_2 are constants.

The initial true and estimated states are taken as $\theta_0 = 0$ and $\tilde{\theta}_{0|0} = \pi$, respectively, while the initial variance is taken as $\mathbf{P}_{0|0} = 2$. We assign $d_1 = 0.1$, $d_2 = 0.15$, $\mathbf{Q}_k = 0.1$, and $\mathbf{R}_k = diag(0.2, 0.2)$. The simulation is performed for 200 time-steps and angular RMSEs are computed by implementing 1000 Monte-Carlo simulations.

Fig. 6.2 shows the angular RMSE plots and Table-6.2 shows the average angular RMSEs of the LRSPF, GRSPF, and the existing filters using different numbers of particles. The figure and table together indicate that RMSE is significantly reduced for both the LRSPF and GRSPF, as compared to the existing choices of proposal densities, *viz* PF, UPF, CPF, and GHPF. However, it is also interesting to note that the existing circular filter CF outperforms the other existing filters.

The RMSE of the GRSPF is higher than the LRSPF, as expected. However, even *global* sampling using wrapped normal distribution as a proposal (*i.e.* the GRSPF) leads to a better estimation accuracy than Gaussian density proposal filters using *local* sampling, even as local sampling leads to a significantly higher computational cost.

Relative computational times of different filters are listed in Table 6.3. Thus, the computational time for the GRSPF is lower than all the algorithms which require local sampling, while it still delivers a superior estimation performance compared to PF, UPF, CPF, and GHPF. CF is not a particle filtering algorithm and has a constant computational cost as well as constant RMSE, relative to the number of particles. Note that global sampling in UPF, CPF, and GHPF leads to poorer accuracy than that using local sampling, and hence, the results of the global sampling for these algorithms are omitted. LRSPF further increases the accuracy of our algorithm, at the expense of somewhat higher computational cost.

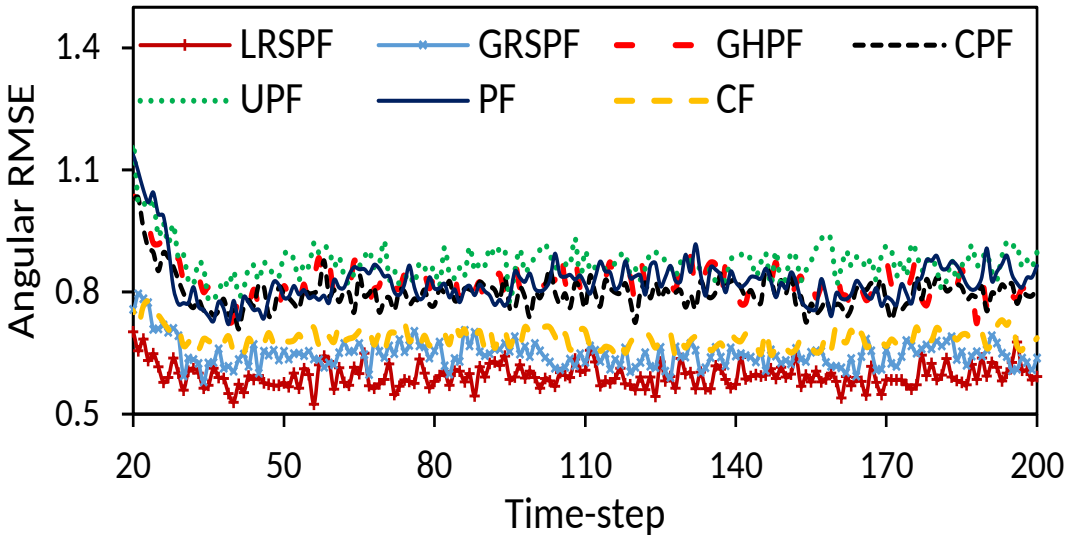


Figure 6.2: Problem 1: Angular RMSE plot with time for the proposed and the existing filters.

Table 6.2: Problem 1: Average angular RMSEs obtained for the proposed filter and the existing filters with varying number of particles.

Filters	10 particles	50 particles	100 particles
PF	0.838	0.827	0.809
UPF	0.937	0.927	0.883
CPF	0.865	0.855	0.810
GHPF	0.911	0.900	0.830
CF	0.684	0.684	0.684
GRSPF	0.698	0.679	0.663
LRSPF	0.674	0.655	0.595

Table 6.3: Problem 1: Relative computational times for the proposed filter and the existing filters with varying number of particles.

Filters	10 particles	50 particles	100 particles
PF	1	1	1
UPF	2.19	2.13	2.02
CPF	1.93	1.83	1.71
GHPF	1.85	1.83	1.72
CF	4.53	1.22	0.63
GRSPF	1.47	1.44	1.33
LRSPF	1.83	1.85	1.74

6.4.2 Problem 2

The second problem considered is a general multivariate nonlinear angular estimation problem [97]. In this problem, the state dynamics are of oscillatory nature, while the measurement equation

Table 6.4: Problem 2: Average angular RMSEs obtained for the proposed filter and the existing filters with varying number of particles.

Filters	10-particles			50-particles			100-particles		
	State 1	State 2	State 3	State 1	State 2	State 3	State 1	State 2	State 3
PF	0.768	0.748	0.761	0.666	0.648	0.660	0.641	0.625	0.633
UPF	0.681	0.665	0.672	0.621	0.604	0.610	0.607	0.591	0.601
CPF	0.663	0.646	0.658	0.613	0.599	0.605	0.603	0.587	0.598
GHPF	0.668	0.649	0.656	0.615	0.599	0.605	0.604	0.588	0.597
GRSPF	0.580	0.564	0.589	0.539	0.521	0.543	0.524	0.508	0.522
LRSPF	0.553	0.535	0.542	0.509	0.490	0.500	0.500	0.487	0.495

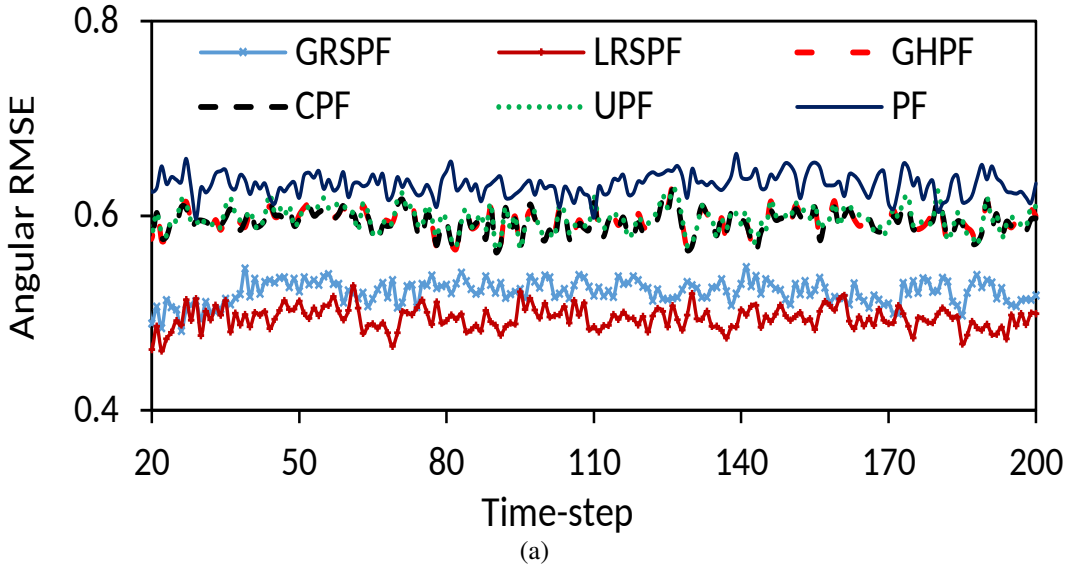
is a monotone increasing function of arguments (*e.g.* a positive quadratic form or its positive square root). The state-space model of this problem can be written as [138, 158],

$$\boldsymbol{\theta}_k = |2\cos(\boldsymbol{\theta}_{k-1})| + \eta_{k-1}, \tag{6.39}$$

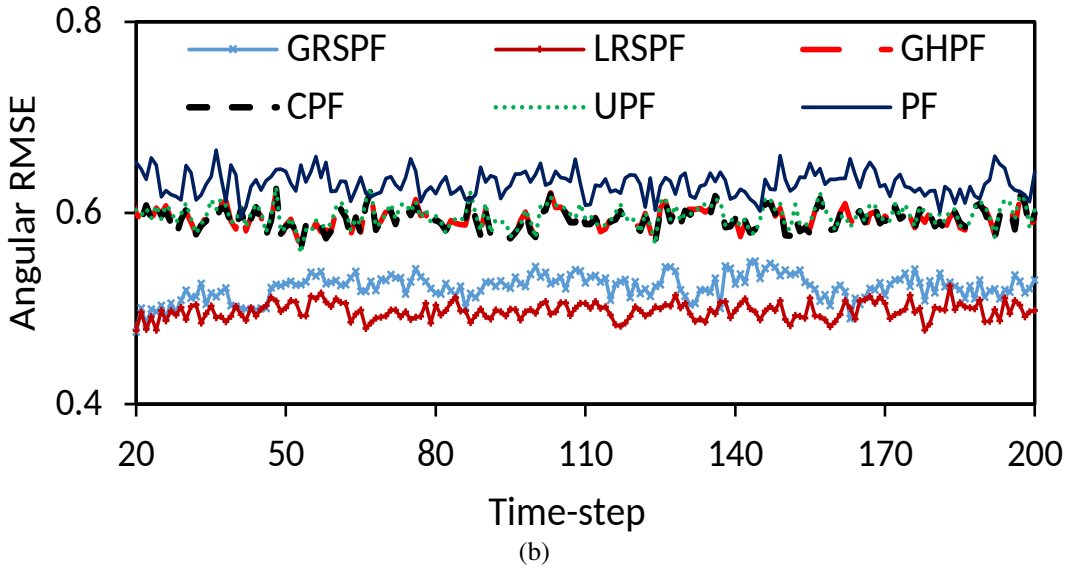
$$\mathbf{y}_k = \sqrt{(1+\boldsymbol{\theta}_k^T\boldsymbol{\theta}_k)} + \vartheta_k. \tag{6.40}$$

We consider a three-dimensional system ($\boldsymbol{\theta}_k \in \mathbb{D}^3$, $\mathbf{y}_k \in \mathbb{D}$) and assign the initial true and estimated states as $\boldsymbol{\theta}_0 = [0, -\pi, \pi]^T$ and $\tilde{\boldsymbol{\theta}}_{0|0} = [-\pi, \pi, \pi/2]^T$, respectively, while initial error covariance is taken as $\mathbf{P}_{0|0} = 2\mathbf{I}_n$. The noise covariances are assigned as $\mathbf{Q}_k = diag([0.05, 0.05, 0.05])$ and $\mathbf{R}_k = 0.1$. The states are estimated for 200 time-steps and the results are evaluated by performing 1000 Monte-Carlo runs.

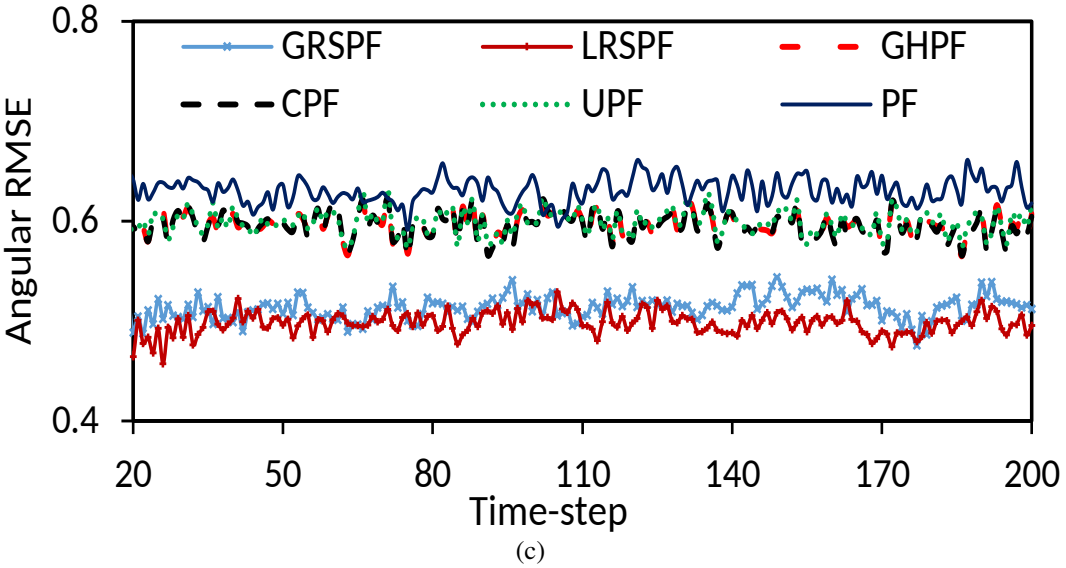
Fig. 6.3 shows the angular RMSE plots of the three states obtained using the LRSPF, GRSPF, and the existing filters. Moreover, Table-6.4 shows the average angular RMSEs obtained using each filter for varying number of particles. Note that the figures and table do not include the results for CF, as this problem is multivariate. As in the case of problem 1, LRSPF and GRSPF yield lower average angular RMSE than all the other filters examined. The order of computational times of all filters was observed similar to the order reported for the first problem. In this case, as well, GRSPF gives better accuracy than existing Gaussian proposal filters at a lower computational cost.



(a)



(b)



(c)

Figure 6.3: Problem 2: Angular RMSE plot with time for the proposed and the existing filters: (a) State 1, (b) State 2, and (2) State 3.

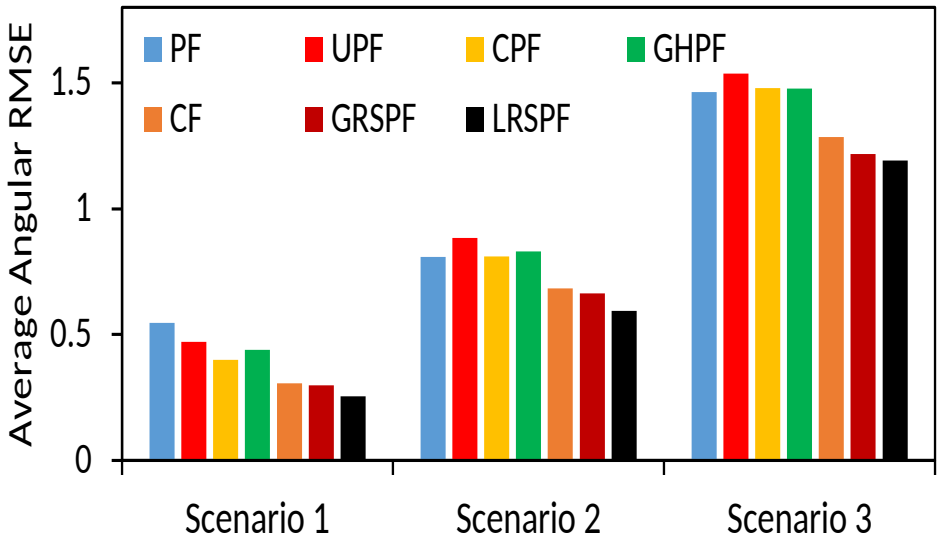


Figure 6.4: Problem 1: Comparison of the average angular RMSEs for the proposed RSPF and the existing filters under three different scenarios.

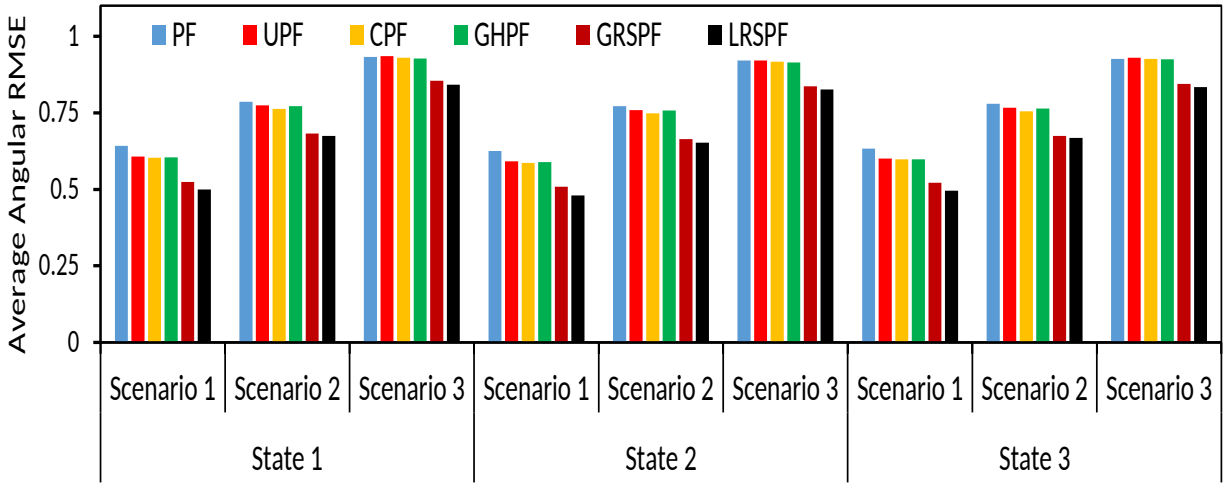


Figure 6.5: Problem 2: Comparison of the average angular RMSEs for the proposed RSPF and the existing filters under three different scenarios.

6.4.3 Effect of noise variations

We further study the effect of varying process and measurement noise parameters on the performance of the RSPF. In this regard, we define three different scenarios by varying the process and measurement noise covariances $\mathbf{Q}_k$ and $\mathbf{R}_k$. For the first problem, the three different scenarios are defined as: Scenario 1: $\mathbf{Q}_k = 0.05$, and $\mathbf{R}_k = diag(0.02, 0.02)$, Scenario 2: $\mathbf{Q}_k = 0.1$, and $\mathbf{R}_k = diag(0.2, 0.2)$, and Scenario 3: $\mathbf{Q}_k = 0.5$, and $\mathbf{R}_k = diag(1, 1)$. Similarly, we define the three scenarios for the second problem as: Scenario 1: $\mathbf{Q}_k = diag([0.05, 0.05, 0.05])$ and $\mathbf{R}_k = 0.1$, Scenario 2: $\mathbf{Q}_k = diag([0.25, 0.25, 0.25])$ and $\mathbf{R}_k = 0.5$, and Scenario 3: $\mathbf{Q}_k = diag([0.5, 0.5, 0.5])$ and $\mathbf{R}_k = 1$. We draw the bar charts of the average angular RMSEs obtained for the proposed and the

existing filters for all three scenarios in Fig. 6.4 for Problem 1 and in Fig. 6.5 for Problem 2. In all the cases, the average angular RMSE is lower for the LRSPF and the GRSPF as compared to the other filters.

6.4.4 Result Discussion

The simulation results infer that the RSPF outperforms the traditional PF, the existing extensions of the PF (such as the UPF, CPF, and GHPF), and the circular filter CF for angular data. Interestingly, the RSPF could outperform all the existing forms of PF at a lower computational demand. We also introduce two forms of the RSPF, abbreviated as LRSPF and GRSPF. They can be used to achieve a different trade-off between the accuracy and computational demand, particularly if the number of particles is high. We also observe that the RSPF outperforms all the existing filters for various noisy environments, which validates the improved accuracy of the proposed method under different practical scenarios.

6.5 Summary

The particle filtering is a popular and widely accepted nonlinear filtering method available in the literature. A crucial determinant of its performance is the choice of proposal density. Most existing particle filters use the Gaussian proposal and tend to perform poorly for the estimation of angular quantities on a restricted domain. The RSPF uses wrapped normal distribution instead of the ordinary normal distribution as the proposal density. Subsequently, it closely represents the proposal density for angular data. Further, we propose to use (i) conditional density approximation similar to the Gaussian filtering case to derive posterior densities and (ii) use Rogers-Szegő quadrature rule along with the product rule for approximating the integrals involved. We compare the performance of two variants of the new filter (with local sampling and with global sampling) with existing Gaussian filters. In two different simulation examples, the proposed filter is shown to outperform Gaussian proposal filters comprehensively. We show that even a global sampling variant of our filter is more accurate than the local sampling versions of existing Gaussian filters, even though global sampling leads to a significantly reduced computational cost. Local sampling version of the RSPF can lead to somewhat increased accuracy at the cost of a higher computational load. Additionally, the product rule suffers from the curse of dimensionality problem [61].

However, the practitioners may replace the product rule with the Smolyak rule [61] and adaptive sparse-grid method [62] to partially address this problem.

Chapter 7

Discussion and Conclusion

7.1 Discussion

The filtering algorithms provide strong stochastic mathematical tools for handling the core computational problems that appear in tracking, modeling, monitoring, diagnosis, prognosis, *etc*. As a result, the filtering algorithms are widely applicable in the fields of target tracking, financial modeling and monitoring, biomedical diagnosis and monitoring, remaining useful-life prediction of industrial equipment, fault diagnosis and prognosis, *etc*.

Despite the widespread applications of the filtering algorithms, the literature lacks an optimal filtering method for nonlinear dynamical systems. Regardless, a variety of nonlinear filters are available with varying accuracy levels and requiring different computational budget. The existing nonlinear filters can be, in general, broadly categorized into Gaussian filters and particle filters. Thankfully, ignoring the limitations on the computational budget, the existing nonlinear filters are adequately accurate for general real-life applications as long as the measurements do not consist of serious irregularities. Nevertheless, the presence of measurement irregularities can severely damage the accuracy, limiting the practical applicability of filtering algorithms.

This thesis is motivated to develop advanced nonlinear filtering algorithms to handle various measurement irregularities. The particular irregularities of interest include time-delayed measurements, non-Gaussian measurement noises, cyber-attacked measurements, and range-limited angular data.

The detailed literature review introduced some filtering methods to handle the above-mentioned measurement irregularities. Nonetheless, they suffer from various limitations. For example, the

existing delay filters require a large number of delay probabilities, which are often ambiguously assigned during the filtering. Thus, the possibility of a wrong selection of delay probabilities increases with the increasing number of the prerequisite delay probabilities. Similarly, the existing filters can handle only one measurement irregularity and they eventually underperform or fail if multiple measurement irregularities appear simultaneously. Furthermore, existing filters for handling the cyber-attacked measurements are designed to handle only one of the FDI, TAM, and DoS attacks, while their simultaneous occurrences can not be denied. Finally, the existing filters mostly consider the data spread over infinite-range and underperform for finite-range data, such as angular data. The research is ongoing to mitigate these limitations of the existing methods.

7.1.1 Limitations

Below are some major limitations of this thesis that have been highlighted.

- Chapter 3 involves the use of two probabilistic pieces of information to handle larger delays (computing this probabilistic information in formerly is a major restraint) with a requirement of high memory storage demands.
- Chapter 4 presented the idea of dealing with multiple measurement irregularities, such as delay and non-Gaussian noises, simultaneously. Hence, this approach faces significant challenges, including high memory storage requirements and substantial computational demands in comparison to the conventional Gaussian filter.
- Chapter 5 devised a filtering algorithm that mandates preprocessing of data to enable the compensation of an FDI attack using a suitable Gaussian PDF. Thus, this preprocessing relies on several assumptions and approximations, which can potentially negatively impact accuracy.
- Chapter 6 presented a filtering technique that is susceptible to the curse of dimensionality issue. As the number of dimensions increases, the computational demands of this method grow exponentially.

7.2 Conclusion

- The accuracy of the existing filters for delayed measurement is improved by introducing an advanced Gaussian filtering method, which reduces the number of prerequisite delay probabilities. The number of the prerequisite delay probabilities is reduced by utilizing a geometric random variable in place of the traditionally used sequence of Bernoulli random variables.

- An advanced Gaussian filtering method is designed for handling the delayed measurements and non-Gaussian measurement noises simultaneously. To handle the delayed measurements, the proposed method identifies the unknown delay stochastically and restructures the traditional Gaussian filtering method accordingly. Moreover, for handling the non-Gaussian measurement noises, the proposed method is designed under the MC criterion in place of the traditionally used MMSE criterion.

- The traditional Gaussian filtering method is redesigned for handling cyber-attacked measurements. Interestingly, the proposed method can simultaneously handle multiple forms of attack, including the FDI, TAM, and DoS attacks. To design the filter, the traditional measurement model is reformulated for incorporating the possibilities of appearing the different forms of attack. Subsequently, the traditional Gaussian filtering method is re-derived for the modified measurement model.

- The traditional particle filtering method is redesigned for handling angular data, which is a class of range-limited data. In this regard, the proposal density is taken as wrapped Gaussian. Thereon, an advanced numerical approximation method is introduced for approximating the numerical integrals that appear in characterizing the wrapped Gaussian approximated proposal density.

- The improved accuracies of the above discussed filters are validated through the simulation results obtained for two simulation problems.

7.3 Future Research Scope

The contributions of this thesis may be extended further in several aspects to further improve the accuracy, reliability, and applicability. Some of the possible extensions are highlighted below.

- The existing delay filters, including the one proposed in Chapter 3, require a set of probabilistic information about the delay. The future research scope of this work may be to develop an advanced filtering method, which derives the PDF of delay, and then, handles the delay accordingly.

- This thesis, particularly Chapter 4, introduced the concept of simultaneously handling more than one measurement irregularities, including the delay and non-Gaussian noises. The future research scope may be to handle more than two irregularities simultaneously.

- The filtering algorithm designed in Chapter 5 requires some preprocessing of data to ensure that the FDI attack can be compensated with an appropriate Gaussian PDF. Such preprocessing ultimately requires various assumptions and approximations, which often influence the accuracy adversely. In the future research, the proposed method can be extended further to relax the need of such data preprocessing.

- The filtering method introduced in Chapter 6 suffers from the curse of dimensionality problem, as its computational demand increases exponentially with increasing dimension. The future research objective may be to reduce the computational demand without harming the accuracy considerably.

- Depending on the application, state dynamics may be complex or unknown. Therefore, machine learning or deep learning techniques can be useful to identify these dynamics, and the above solutions can also be applied to unknown dynamics.

Appendices

Appendix A

Simplifying Eq. (4.25) in terms of Eq. (4.26)

Let us expand $\left(\mathbf{W}_{\tilde{g},\tilde{d}}^{T}\mathbb{S}_{\tilde{g},\tilde{d}}^{t-1}\mathbf{W}_{\tilde{g},\tilde{d}}\right)^{-1}$ and $\left(\mathbf{W}_{\tilde{g},\tilde{d}}^{T}\mathbb{S}_{\tilde{g},\tilde{d}}^{t-1}\mathbf{D}_{\tilde{g},\tilde{d}}\right)$. To expand $\left(\mathbf{W}_{\tilde{g},\tilde{d}}^{T}\mathbb{S}_{\tilde{g},\tilde{d}}^{t-1}\mathbf{W}_{\tilde{g},\tilde{d}}\right)^{-1}$, we rewrite $\mathbf{W}_{\tilde{g},\tilde{d}}$ given in Eq. (4.20) by substituting $\Omega_{\tilde{g},\tilde{d}}$ from Eq. (4.18). Then, given $\mathbb{S}_{\tilde{g},\tilde{d}} = diag\left(\mathbb{S}_{x,(\tilde{g},\tilde{d})}, \mathbb{S}_{z,(\tilde{g},\tilde{d})}\right)$ and the resulting $\mathbf{W}_{\tilde{g},\tilde{d}}$ expression, we obtain $\left(\mathbf{W}_{\tilde{g},\tilde{d}}^{T}\mathbb{S}_{\tilde{g},\tilde{d}}^{t-1}\mathbf{W}_{\tilde{g},\tilde{d}}\right)^{-1}$ in the form of Eq. (A.1).

$$\left(\mathbf{W}_{\tilde{g},\tilde{d}}^{T}\mathbb{S}_{\tilde{g},\tilde{d}}^{t-1}\mathbf{W}_{\tilde{g},\tilde{d}}\right)^{-1} = \left((\Omega_{\tilde{g},\tilde{d}}^{nT})^{-1}\mathbb{S}_{x,(\tilde{g},\tilde{d})}^{t-1}(\Omega_{\tilde{g},\tilde{d}}^{n})^{-1} + \mathbb{H}_{\tilde{g},\tilde{d}}^{T}(\Omega_{\tilde{g},\tilde{d}}^{rT})^{-1}\mathbb{S}_{z,(\tilde{g},\tilde{d})}^{t-1}(\Omega_{\tilde{g},\tilde{d}}^{r})^{-1}\mathbb{H}_{\tilde{g},\tilde{d}}\right)^{-1}. \quad \text{(A.1)}$$

To further simplify Eq. (A.1), let us consider the following notations

$$\begin{cases} A = (\Omega_{\tilde{g},\tilde{d}}^{nT})^{-1}\mathbb{S}_{x,(\tilde{g},\tilde{d})}^{t-1}(\Omega_{\tilde{g},\tilde{d}}^{n})^{-1}, \;\; C = \mathbb{H}_{\tilde{g},\tilde{d}}^{T} \\ D = (\Omega_{\tilde{g},\tilde{d}}^{rT})^{-1}\mathbb{S}_{z,(\tilde{g},\tilde{d})}^{t-1}(\Omega_{\tilde{g},\tilde{d}}^{r})^{-1}, \;\; E = \mathbb{H}_{\tilde{g},\tilde{d}} \end{cases} \quad \text{(A.2)}$$

Considering Woodbury matrix identity, $(A+CDE)^{-1} = A^{-1} - A^{-1}C(D^{-1}+EA^{-1}C)^{-1}EA^{-1}$, which gives $\left(\mathbf{W}_{\tilde{g},\tilde{d}}^{T}\mathbb{S}_{\tilde{g},\tilde{d}}^{t-1}\mathbf{W}_{\tilde{g},\tilde{d}}\right)^{-1}$ expression in the form of Eq. (A.3).

$$\begin{aligned} \left(\mathbf{W}_{\tilde{g},\tilde{d}}^{T}\mathbb{S}_{\tilde{g},\tilde{d}}^{t-1}\mathbf{W}_{\tilde{g},\tilde{d}}\right)^{-1} = {} & \Omega_{\tilde{g},\tilde{d}}^{n}(\mathbb{S}_{x,(\tilde{g},\tilde{d})}^{t-1})^{-1}\Omega_{\tilde{g},\tilde{d}}^{nT} - \Omega_{\tilde{g},\tilde{d}}^{n}(\mathbb{S}_{x,(\tilde{g},\tilde{d})}^{t-1})^{-1}\Omega_{\tilde{g},\tilde{d}}^{nT}\mathbb{H}_{\tilde{g},\tilde{d}}^{T}\Big(\Omega_{\tilde{g},\tilde{d}}^{r}(\mathbb{S}_{z,(\tilde{g},\tilde{d})}^{t-1})^{-1}\Omega_{\tilde{g},\tilde{d}}^{rT} \\ & + \mathbb{H}_{\tilde{g},\tilde{d}}\Omega_{\tilde{g},\tilde{d}}^{n}(\mathbb{S}_{x,(\tilde{g},\tilde{d})}^{t-1})^{-1}\Omega_{\tilde{g},\tilde{d}}^{nT}\mathbb{H}_{\tilde{g},\tilde{d}}^{T}\Big)^{-1}\mathbb{H}_{\tilde{g},\tilde{d}}\Omega_{\tilde{g},\tilde{d}}^{n}(\mathbb{S}_{x,(\tilde{g},\tilde{d})}^{t-1})^{-1}\Omega_{\tilde{g},\tilde{d}}^{nT}. \end{aligned} \quad \text{(A.3)}$$

To obtain $\mathbf{W}_{\tilde{g},\tilde{d}}^{T}\mathbb{S}_{\tilde{g},\tilde{d}}^{t-1}\mathbf{D}_{\tilde{g},\tilde{d}}$, let us rewrite $\mathbf{D}_{\tilde{g},\tilde{d}}$ by substituting $\Omega_{\tilde{g},\tilde{d}}$ from Eq. (4.18) into Eq.

(4.19) and expressed as

$$\mathbf{D}_{\tilde{g},\tilde{d}} = \begin{bmatrix} (\Omega^{n}_{\tilde{g},\tilde{d}})^{-1}\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k-1} \\ (\Omega^{n}_{\tilde{g},\tilde{d}})^{-1}\left(\mathbf{z}_k - \hat{\mathbf{z}}_{\tilde{g},\tilde{d}|k-1} + \mathbb{H}_{\tilde{g},\tilde{d}}\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k-1}\right) \end{bmatrix}. \tag{A.4}$$

Then, for the resulting expression of $\mathbf{D}_{\tilde{g},\tilde{d}}$, $\mathbf{W}_{\tilde{g},\tilde{d}}$, and $\mathbb{S}^{t-1}_{\tilde{g},\tilde{d}}$, we get $\mathbf{W}^{T}_{\tilde{g},\tilde{d}}\mathbb{S}^{t-1}_{\tilde{g},\tilde{d}}\mathbf{D}_{\tilde{g},\tilde{d}}$ in the form of Eq. (A.5).

$$\begin{aligned}\mathbf{W}^{T}_{\tilde{g},\tilde{d}}\mathbb{S}^{t-1}_{\tilde{g},\tilde{d}}\mathbf{D}_{\tilde{g},\tilde{d}} =& (\Omega^{nT}_{\tilde{g},\tilde{d}})^{-1}\mathbb{S}^{t-1}_{x,(\tilde{g},\tilde{d})}(\Omega^{n}_{\tilde{g},\tilde{d}})^{-1}\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k-1} + \mathbb{H}^{T}_{\tilde{g},\tilde{d}}(\Omega^{rT}_{\tilde{g},\tilde{d}})^{-1}\mathbb{S}^{t-1}_{z,(\tilde{g},\tilde{d})}(\Omega^{r}_{\tilde{g},\tilde{d}})^{-1}\left(\mathbf{z}_k \right. \\ & \left. - \hat{\mathbf{z}}_{\tilde{g},\tilde{d}|k-1} + \mathbb{H}_{\tilde{g},\tilde{d}}\hat{\mathbf{x}}_{\tilde{g},\tilde{d}|k-1}\right).\end{aligned} \tag{A.5}$$

To this end, substituting $\left(\mathbf{W}^{T}_{\tilde{g},\tilde{d}}\mathbb{S}^{t-1}_{\tilde{g},\tilde{d}}\mathbf{W}_{\tilde{g},\tilde{d}}\right)^{-1}$ and $\left(\mathbf{W}^{T}_{\tilde{g},\tilde{d}}\mathbb{S}^{t-1}_{\tilde{g},\tilde{d}}\mathbf{D}_{\tilde{g},\tilde{d}}\right)$ from Eqs. (A.3) and (A.5), respectively into Eq. (4.25), we obtain the desired $\hat{\mathbf{x}}^{t}_{\tilde{g},\tilde{d}|k}$ in the form of Eq. (4.26) for $\mathbf{K}^{t-1}_{\tilde{g},\tilde{d}|k}$, $\bar{\mathbf{P}}^{t-1}_{\tilde{g},\tilde{d}|k-1}$, and $\bar{\mathbf{R}}^{t-1}_{\tilde{g},\tilde{d}|k}$ given in Eqs. (4.27), (4.28), and (4.29), respectively.

Appendix B

Analytical Steps of Filtering

Sample points and Weights:

Compute the deterministic sample points and weights using numerical approximation method: $\boldsymbol{\Upsilon}_i$ and $W_i \; \forall i \in \{1,2,\cdots,N_s\}$.

Prediction:

- Determine the Cholesky decomposition of initial error covariance

  $\mathbf{P}_{k-1|k-1} = \boldsymbol{\Sigma}_{k-1|k-1}\boldsymbol{\Sigma}^T_{k-1|k-1}$,

  where $\boldsymbol{\Sigma}_{k-1|k-1} = chol(\mathbf{P}_{k-1|k-1}, lower)$

- Compute the transformed sampling points

  $\zeta_{i,k-1|k-1} = \boldsymbol{\Sigma}_{k-1|k-1}\boldsymbol{\Upsilon}_i + \hat{\mathbf{x}}_{k-1|k-1}$

- Propagate $\zeta_{i,k-1|k-1}$ through process model

  $\zeta^*_{i,k-1|k-1} = \phi_{k-1}(\zeta_{i,k-1|k-1})$.

- Estimate the predicated mean

  $\hat{\mathbf{x}}_{k|k-1} = \sum_{i=1}^{N_s} W_i \zeta^*_{i,k-1|k-1}$.

- Compute the predicated error covariance

  $\mathbf{P}_{k|k-1} = \sum_{ji=1}^{N_s} W_i \zeta^*_{i,k-1|k-1}\zeta^{*T}_{i,k-1|k-1} - \hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{x}}^T_{k|k-1} + \mathbf{Q}_k$.

Update:

- Determine the Cholesky decomposition of predicted error covariance

  $\mathbf{P}_{k-1|k} = \mathbf{\Sigma}_{k-1|k}\mathbf{\Sigma}_{k-1|k}^{T}$,

  where $\mathbf{\Sigma}_{k-1|k} = chol(\mathbf{P}_{k-1|k}, lower)$

- Compute the transformed sample points

  $\zeta_{i,k|k-1} = \mathbf{\Sigma}_{k|k-1}\mathbf{\Upsilon}_i + \hat{\mathbf{x}}_{k|k-1}$

- Propagate $\zeta_{i,k|k-1}$ through measurement model

  $\zeta^{*}_{i,k|k-1} = \Psi_k(\zeta_{i,k|k-1})$

- Compute the predicted measurement

  $\hat{\mathbf{z}}_{k|k-1} = \sum_{j=1}^{N_s} W_i \zeta^{*}_{i,k|k-1}$

- Compute the innovation error covariance

  $\mathbf{P}^{\mathbf{zz}}_{k|k-1} = \sum_{i=1}^{N_s} W_i \zeta^{*}_{i,k|k-1}\zeta^{*T}_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1}\hat{\mathbf{z}}^{T}_{k|k-1} + \mathbf{R}_k$

- Compute the cross-covariance

  $\mathbf{P}^{\mathbf{xz}}_{k|k-1} = \sum_{i=1}^{N_s} W_i \zeta_{i,k|k-1}\zeta^{*T}_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{z}}^{T}_{k|k-1}$

- Determine the Kalman gain

  $\mathbf{K}_k = \mathbf{P}^{\mathbf{xz}}_{k|k-1}(\mathbf{P}^{\mathbf{zz}}_{k|k-1})^{-1}$

- Compute the updated estimate of state

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

- Compute the updated error covariance of state

  $\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{P}^{\mathbf{zz}}_{k|k-1}\mathbf{K}_k^{T}$

Bibliography

[1] Francisco J Montáns, Francisco Chinesta, Rafael Gómez-Bombarelli, and J Nathan Kutz. Data-driven modeling and learning in science and engineering. *Comptes Rendus Mécanique*, 347(11):845–855, 2019.

[2] Simon S Haykin. *Kalman filtering and neural networks*, volume 284. Wiley Online Library, 2001.

[3] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[4] Andrew H Jazwinski. *Stochastic processes and filtering theory*. Courier Corporation, 2007.

[5] Rajiv Ramaswami, Kumar Sivarajan, and Galen Sasaki. *Optical networks: a practical perspective*. Morgan Kaufmann, 2009.

[6] Yun Fei, Tao Meng, and Zhonghe Jin. Nano satellite attitude determination with randomly delayed measurements. *Acta Astronautica*, 185:319–332, 2021.

[7] Kebina Manandhar, Xiaojun Cao, Fei Hu, and Yao Liu. Detection of faults and attacks including false data injection attack in smart grid using Kalman filter. *IEEE Transactions on Control of Network Systems*, 1(4):370–379, 2014.

[8] Omid Sayadi and Mohammad Bagher Shamsollahi. Ecg denoising and compression using a modified extended Kalman filter structure. *IEEE Transactions on Biomedical Engineering*, 55(9):2240–2248, 2008.

[9] SY Chen. Kalman filter for robot vision: a survey. *IEEE Transactions on Industrial Electronics*, 59(11):4409–4420, 2011.

[10] Rohit Rana, Vijyant Agarwal, Prerna Gaur, and Harish Parthasarathy. Design of optimal ukf state observer–controller for stochastic dynamical systems. *IEEE Transactions on Industry Applications*, 57(2):1840–1859, 2020.

[11] Xiaolin Ning, Fan Wang, and Jiancheng Fang. An implicit ukf for satellite stellar refraction navigation system. *IEEE Transactions on Aerospace and Electronic Systems*, 53(3):1489–1503, 2017.

[12] Ramaprasad Bhar. *Stochastic filtering with applications in finance*. World Scientific, 2010.

[13] Federico Cassola and Massimiliano Burlando. Wind speed and wind energy forecast through Kalman filtering of numerical weather prediction model output. *Applied energy*, 99:154–166, 2012.

[14] Antonio Gomez-Exposito, Jose A Rosendo-Macias, and Miguel A Gonzalez-Cagigal. Monitoring and tracking the evolution of a viral epidemic through nonlinear Kalman filtering: Application to the covid-19 case. *IEEE Journal of Biomedical and Health Informatics*, 26(4):1441–1452, 2021.

[15] Arthur Gelb et al. *Applied optimal estimation*. MIT press, 1974.

[16] ROBERTG BROWN and PATRICKYC HWANG. Introduction to random signals and applied Kalman filtering (book). *New York, John Wiley & Sons, Inc., 1992. 512*, 1992.

[17] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluid Engineering*, 82,:35—-45, 1960.

[18] Felix Govaers. *Introduction and Implementations of the Kalman Filter*. BoD–Books on Demand, 2019.

[19] Marco Huber. *Nonlinear Gaussian Filtering: Theory, Algorithms, and Applications*, volume 19. KIT Scientific Publishing, 2015.

[20] Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140, pages 107–113. IET, 1993.

[21] Zhe Chen et al. Bayesian filtering: From Kalman filters to particle filters, and beyond. *Statistics*, 182(1):1–69, 2003.

[22] Abhinoy Kumar Singh. Major development under Gaussian filtering since unscented Kalman filter. *IEEE/CAA Journal of Automatica Sinica*, In press.

[23] Shovan Bhaumik and Paresh Date. *Nonlinear estimation: methods and applications with deterministic Sample Points*. Chapman and Hall/CRC, 2019.

[24] Huazhen Fang, Ning Tian, Yebin Wang, MengChu Zhou, and Mulugeta A Haile. Nonlinear Bayesian estimation: From Kalman filtering to a broader horizon. *IEEE/CAA Journal of Automatica Sinica*, 5(2):401–417, 2018.

[25] Bin Jia and Ming Xin. *Grid-based nonlinear estimation and its applications*. CRC Press, 2019.

[26] Brian DO Anderson and John B Moore. *Optimal filtering*. Courier Corporation, 2012.

[27] Michael Athans, Richard Wishner, and Anthony Bertolini. Suboptimal state estimation for continuous-time nonlinear systems from discrete noisy measurements. *IEEE Transactions on Automatic Control*, 13(5):504–514, 1968.

[28] Manoranjan Majji, John L Junkins, and James D Turner. A high order method for estimation of dynamic systems. *The Journal of the Astronautical Sciences*, 56(3):401–440, 2008.

[29] Taek Song and Jason Speyer. A stochastic analysis of a modified gain extended Kalman filter with applications to estimation with bearings only measurements. *IEEE Transactions on Automatic Control*, 30(10):940–949, 1985.

[30] Norman Morrison. *Tracking filter engineering: the Gauss-Newton and polynomial filters*, volume 23. IET, 2012.

[31] PK Dash, KR Krishnanand, and M Padhee. Fast recursive gauss–newton adaptive filter for the estimation of power system frequency and harmonics in a noisy environment. *IET generation, transmission & distribution*, 5(12):1277–1289, 2011.

[32] Roaldje Nadjiasngar, Stephen Middleton, and Michael Inggs. Doppler-only tracking with the recursive gauss-newton filter. 2012.

[33] Roaldje Nadjiasngar and Michael Inggs. Gauss–newton filtering incorporating levenberg–marquardt methods for tracking. *Digital Signal Processing*, 23(5):1662–1667, 2013.

[34] Simon J Julier and Jeffrey K Uhlmann. New extension of the Kalman filter to nonlinear systems. In *Signal Processing, Sensor Fusion, and Target Recognition VI*, volume 3068, pages 182–194. Int. Soc. Opt. Phot., 1997.

[35] Simon J Julier and Jeffrey K Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[36] Simon J Julier. The scaled unscented transformation. In *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, volume 6, pages 4555–4559. IEEE, 2002.

[37] Rudolph Van Der Merwe and Eric A Wan. The square-root unscented kalman filter for state and parameter-estimation. In *2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221)*, volume 6, pages 3461–3464. IEEE, 2001.

[38] Zhe Jiang, Qi Song, Yuqing He, and Jianda Han. A novel adaptive unscented Kalman filter for nonlinear estimation. In *2007 46th IEEE Conference on Decision and Control*, pages 4293–4298. IEEE, 2007.

[39] Lubin Chang, Baiqing Hu, An Li, and Fangjun Qin. Transformed unscented Kalman filter. *IEEE Transactions on Automatic Control*, 58(1):252–257, 2012.

[40] Li Wan-Chun, Wei Ping, and Xiao Xian-Ci. A novel simplex unscented transform and filter. In *2007 International Symposium on Communications and Information Technologies*, pages 926–931. IEEE, 2007.

[41] Juan G Castrejón Lozano, Luis R Garca Carrillo, Alejandro Dzul, and Rogelio Lozano. Spherical simplex sigma-point Kalman filters: a comparison in the inertial navigation of a terrestrial vehicle. In *2008 American Control Conference*, pages 3536–3541. IEEE, 2008.

[42] Lubin Chang, Baiqing Hu, Guobin Chang, and An Li. Marginalised iterated unscented Kalman filter. *IET Control Theory & Applications*, 6(6):847–854, 2012.

[43] S Bhaumik, S Sadhu, and TK Ghoshal. Risk-sensitive formulation of unscented Kalman filter. *IET control theory & applications*, 3(4):375–382, 2009.

[44] Rahul Radhakrishnan, Ajay Yadav, Paresh Date, and Shovan Bhaumik. A new method for generating sigma points and weights for nonlinear filtering. *IEEE Control Systems Letters*, 2(3):519–524, 2018.

[45] Menatalla Shehab El Din, Ala A Hussein, and Mamoun F Abdel-Hafez. Improved battery soc estimation accuracy using a modified ukf with an adaptive cell model under real ev operating conditions. *IEEE Transactions on Transportation Electrification*, 4(2):408–417, 2018.

[46] Junjian Qi, Kai Sun, Jianhui Wang, and Hui Liu. Dynamic state estimation for multi-machine power system by unscented Kalman filter with enhanced numerical stability. *IEEE Transactions on Smart Grid*, 9(2):1184–1196, 2016.

[47] Mahmoud El-Gohary and James McNames. Human joint angle estimation with inertial sensors and validation with a robot arm. *IEEE Transactions on Biomedical Engineering*, 62(7):1759–1767, 2015.

[48] Ienkaran Arasaratnam and Simon Haykin. Cubature Kalman filters. *IEEE Transaction on Automatic Control*, 54(6):1254–1269, 2009.

[49] Shovan Bhaumik et al. Cubature quadrature Kalman filter. *IET Signal Processing*, 7(7):533–541, 2013.

[50] Bin Jia, Ming Xin, and Yang Cheng. High-degree cubature Kalman filter. *Automatica*, 49(2):510–518, 2013.

[51] Abhinoy Kumar Singh and Shovan Bhaumik. Higher degree cubature quadrature Kalman filter. *International Journal of Control, Automation and Systems*, 13(5):1097–1105, 2015.

[52] Shovan Bhaumik and Swati. Square-root cubature-quadrature Kalman filter. *Asian Journal of Control*, 16(2):617–622, 2014.

[53] Abhinoy Kumar Singh and Shovan Bhaumik. Transformed cubature quadrature Kalman filter. *IET Signal Processing*, 11(9):1095–1103, 2017.

[54] Shiyuan Wang, Jiuchao Feng, and K Tse Chi. Spherical simplex-radial cubature Kalman filter. *IEEE Signal Processing Letters*, 21(1):43–46, 2013.

[55] Zhaoming Li and Wenge Yang. Spherical simplex-radial cubature quadrature Kalman filter. *Journal of Electrical and Computer Engineering*, 2017, 2017.

[56] Abhinoy Kumar Singh. Exponentially fitted cubature Kalman filter with application to oscillatory dynamical systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 67(8):2739–2752, 2020.

[57] Abhinoy Kumar Singh, Mihailo V Rebec, and Ahmad Haidar. Kalman-based calibration algorithm for agamatrix continuous glucose monitoring system. *IEEE Transactions on Control Systems Technology*, 29(3):1257–1267, 2020.

[58] Rahul Radhakrishnan, Abhinoy Kumar Singh, Shovan Bhaumik, and Nutan Kumar Tomar. Quadrature filters for underwater passive bearings-only target tracking. In *2015 Sensor Signal Processing for Defence (SSPD)*, pages 1–5. IEEE, 2015.

[59] Selma Zoljic-Beglerovic, Georg Stettinger, Bernd Luber, and Martin Horn. Railway suspension system fault diagnosis using cubature Kalman filter techniques. *IFAC-PapersOnLine*, 51(24):1330–1335, 2018.

[60] Ienkaran Arasaratnam, Simon Haykin, and Robert J Elliott. Discrete-time nonlinear filtering algorithms using Gauss–Hermite quadrature. *Proceedings of the IEEE*, 95(5):953–977, 2007.

[61] Bin Jia, Ming Xin, and Yang Cheng. Sparse-grid quadrature nonlinear filtering. *Automatica*, 48(2):327–341, 2012.

[62] Abhinoy Kumar Singh, Rahul Radhakrishnan, Shovan Bhaumik, and Paresh Date. Adaptive sparse-grid gauss–hermite filter. *Journal of Computational and Applied Mathematics*, 342:305–316, 2018.

[63] Ienkaran Arasaratnam and Simon Haykin. Square-root quadrature Kalman filtering. *IEEE Transactions on Signal Processing*, 56(6):2589–2593, 2008.

[64] Hermann Singer. Generalized gauss–hermite filtering. *AStA Advances in Statistical Analysis*, 92(2):179–195, 2008.

[65] Rahul Radhakrishnan, Abhinoy Kumar Singh, Shovan Bhaumik, and Nutan Kumar Tomar. Multiple sparse-grid gauss–hermite filtering. *Applied Mathematical Modelling*, 40(7-8):4441–4450, 2016.

[66] Ngai Ming Kwok, Gamini Dissanayake, and Quang Phuc Ha. Bearing-only slam using a sprt based Gaussian sum filter. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1109–1114. IEEE, 2005.

[67] Mahshad Valipour and Luis A Ricardez-Sandoval. Abridged Gaussian sum extended Kalman filter for nonlinear state estimation under non-gaussian process uncertainties. *Computers & Chemical Engineering*, 155:107534, 2021.

[68] Murat Gokce and Mustafa Kuzuoglu. Unscented Kalman filter-aided Gaussian sum filter. *IET Radar, Sonar & Navigation*, 9(5):589–599, 2015.

[69] Pei H Leong, Sanjeev Arulampalam, Tharaka A Lamahewa, and Thushara D Abhayapala. A gaussian-sum based cubature Kalman filter for bearings-only tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 49(2):1161–1176, 2013.

[70] Chen Qian, Chengying Song, Sheng Li, Qingwei Chen, and Jian Guo. Algorithm of Gaussian sum filter based on sgqf for nonlinear non-gaussian models. *International Journal of Control, Automation and Systems*, 19(8):2830–2841, 2021.

[71] Herman K Van Dijk and Teun Kloek. Further experience in Bayesian analysis using monte carlo integration. *Journal of Econometrics*, 14(3):307–328, 1980.

[72] Xiaoyan Fu and Yingmin Jia. An improvement on resampling algorithm of particle filters. *IEEE Transactions on Signal Processing*, 58(10):5414–5420, 2010.

[73] Matthew A Nicely and B Earl Wells. Improved parallel resampling methods for particle filtering. *IEEE Access*, 7:47593–47604, 2019.

[74] Chanin Kuptametee and Nattapol Aunsri. A review of resampling techniques in particle filtering framework. *Measurement*, page 110836, 2022.

[75] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transaction on Signal Processing*, 50(2):174–188, 2002.

[76] Mark R Morelande and Subhash Challa. Manoeuvring target tracking in clutter using particle filters. *IEEE Transactions on Aerospace and Electronic Systems*, 41(1):252–270, 2005.

[77] Mohamed Ahwiadi and Wilson Wang. An adaptive particle filter technique for system state estimation and prognosis. *IEEE Transactions on Instrumentation and Measurement*, 69(9):6756–6765, 2020.

[78] Qiang Miao, Lei Xie, Hengjuan Cui, Wei Liang, and Michael Pecht. Remaining useful life prediction of lithium-ion battery with unscented particle filter technique. *Microelectronics Reliability*, 53(6):805–810, 2013.

[79] Dong Wang, Fangfang Yang, Kwok-Leung Tsui, Qiang Zhou, and Suk Joo Bae. Remaining useful life prediction of lithium-ion batteries based on spherical cubature particle filter. *IEEE Transactions on Instrumentation and Measurement*, 65(6):1282–1291, 2016.

[80] Yan Ma, Yang Chen, Xiuwen Zhou, and Hong Chen. Remaining useful life prediction of lithium-ion battery based on Gauss–Hermite particle filter. *IEEE Transactions on Control Systems Technology*, 27(4):1788–1795, 2018.

[81] Simo Särkkä, Aki Vehtari, and Jouko Lampinen. Rao-blackwellized particle filter for multiple target tracking. *Information Fusion*, 8(1):2–15, 2007.

[82] Ping Li and Visakan Kadirkamanathan. Particle filtering based likelihood ratio approach to fault diagnosis in nonlinear stochastic systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 31(3):337–343, 2001.

[83] Adam M Johansen and Arnaud Doucet. A note on auxiliary particle filters. *Statistics & Probability Letters*, 78(12):1498–1504, 2008.

[84] Jayesh H Kotecha and Petar M Djuric. Gaussian sum particle filtering. *IEEE Transactions on Signal Processing*, 51(10):2602–2612, 2003.

[85] Rong Yang, Yaakov Bar-Shalom, Hong An Jack Huang, and Gee Wah Ng. Ughf for acoustic tracking with state-dependent propagation delay. *IEEE Transactions on Aerospace and Electronic Systems*, 51(3):1747–1761, 2015.

[86] Yanbo Yang, Yan Liang, Quan Pan, Yuemei Qin, and Xiaoxu Wang. Gaussian-consensus filter for nonlinear systems with randomly delayed measurements in sensor networks. *Information Fusion*, 30:91–102, 2016.

[87] Zhao Wang, Yulong Huang, Yonggang Zhang, Guangle Jia, and Jonathon Chambers. An improved kalman filter with adaptive estimate of latency probability. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 67(10):2259–2263, 2019.

[88] Luca Schenato. Kalman filtering for networked control systems with random delay and packet loss. 2006.

[89] Sen Li, You Hu, Lini Zheng, Zhen Li, Xi Chen, Tyrone Fernando, Herbert HC Iu, Qinglin Wang, and Xiangdong Liu. Stochastic event-triggered cubature Kalman filter for power system dynamic state estimation. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 66(9):1552–1556, 2018.

[90] Hongjiu Yang, Hui Li, Yuanqing Xia, and Li Li. Distributed Kalman filtering over sensor networks with transmission delays. *IEEE Transactions on Cybernetics*, 51(11):5511–5521, 2020.

[91] Dominik Sierociuk, Inés Tejado, and Blas M Vinagre. Improved fractional Kalman filter and its application to estimation over lossy networks. *Signal Processing*, 91(3):542–552, 2011.

[92] Aurora Hermoso-Carazo and Josefa Linares-Pérez. Extended and unscented filtering algorithms using one-step randomly delayed observations. *Applied Mathematics and Computation*, 190(2):1375–1393, 2007.

[93] Xiaoxu Wang, Yan Liang, Quan Pan, and Chunhui Zhao. Gaussian filter for nonlinear systems with one-step randomly delayed measurements. *Automatica*, 49(4):976–986, 2013.

[94] Abhinoy Kumar Singh, Shovan Bhaumik, and Paresh Date. Quadrature filters for one-step randomly delayed measurements. *Applied Mathematical Modelling*, 40(19-20):8296–8308, 2016.

[95] A Hermoso-Carazo and J Linares-Pérez. Unscented filtering algorithm using two-step randomly delayed observations in nonlinear systems. *Applied Mathematics and Computation*, 33(9):3705–3717, 2009.

[96] Kai Zhao, Li-Guo Tan, and Shen-Min Song. Gaussian filter for nonlinear networked systems with synchronously correlated noises and one-step randomly delayed measurements and multiple packet dropouts. *IEEE Sensors Journal*, 19(20):9271–9281, 2019.

[97] Abhinoy Kumar Singh, Shovan Bhaumik, and Paresh Date. A modified Bayesian filter for randomly delayed measurements. *IEEE Transactions on Automatic Control*, 62(1):419–424, 2016.

[98] Abhinoy Kumar Singh, Paresh Date, and Shovan Bhaumik. New algorithm for continuous-discrete filtering with randomly delayed measurements. *IET Control Theory & Applications*, 10(17):2298–2305, 2016.

[99] Ramin Esmzad and Reza Mahboobi Esfanjani. Bayesian filter for nonlinear systems with randomly delayed and lost measurements. *Automatica*, 107:36–42, 2019.

[100] Yang Shi and Huazhen Fang. Kalman filter-based identification for systems with randomly missing measurements in a network environment. *International Journal of Control*, 83(3):538–551, 2010.

[101] Y Shi, H Fang, and M Yan. Kalman filter-based adaptive control for networked systems with unknown parameters and randomly missing outputs. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 19(18):1976–1992, 2009.

[102] Liang Hu, Zidong Wang, Izaz Rahman, and Xiaohui Liu. A constrained optimization approach to dynamic state estimation for power systems including PMU and missing measurements. *IEEE Transactions on Control Systems Technology*, 24(2):703–710, 2015.

[103] Roozbeh Razavi-Far, Maryam Farajzadeh-Zanjani, Mehrdad Saif, and Shiladitya Chakrabarti. Correlation clustering imputation for diagnosing attacks and faults with missing power grid data. *IEEE Transaction on Smart Grid*, 11(2):1453–1464, 2019.

[104] Petar M Djuric, Mahesh Vemula, and Mónica F Bugallo. Target tracking by particle filtering in binary sensor networks. *IEEE Transactions on Signal Processing*, 56(6):2229–2238, 2008.

[105] Sumanta Kumar Nanda, Guddu Kumar, Vimal Bhatia, and Abhinoy Kumar Singh. Kalman filtering with delayed measurements in non-Gaussian environments. *IEEE Access*, 9:123231–123244, 2021.

[106] Jun Hu, Zidong Wang, Huijun Gao, and Lampros K Stergioulas. Extended kalman filtering with stochastic nonlinearities and multiple missing measurements. *Automatica*, 48(9):2007–2015, 2012.

[107] Xiaoli Luan, Yan Shi, and Fei Liu. Unscented kalman filtering for greenhouse climate control systems with missing measurement. *International Journal of Innovative Computing, Information and Control*, 8(3):2173–2180, 2012.

[108] Long Xu, Kemao Ma, and Hongxia Fan. Unscented kalman filtering for nonlinear state estimation with correlated noises and missing measurements. *International Journal of Control, Automation and Systems*, 16(3):1011–1020, 2018.

[109] Jun Hu, Jiaxing Li, Yonggui Kao, and Dongyan Chen. Optimal distributed filtering for nonlinear saturated systems with random access protocol and missing measurements: The uncertain probabilities case. *Applied Mathematics and Computation*, 418:126844, 2022.

[110] Amit Kumar Naik, Guddu Kumar, Prabhat Kumar Upadhyay, Paresh Date, and Abhinoy Kumar Singh. Gaussian filtering for simultaneously occurring delayed and missing measurements. *IEEE Access*, 10:100746–100762, 2022.

[111] Jose C Principe. *Information theoretic learning: Renyi's entropy and kernel perspectives*. Springer Science & Business Media, 2010.

[112] Yulong Huang, Yonggang Zhang, and Jonathon A Chambers. A novel kullback–leibler divergence minimization-based adaptive student's t-filter. *IEEE Transaction on Signal Processing*, 67(20):5417–5432, 2019.

[113] Badong Chen, Xi Liu, Haiquan Zhao, and Jose C Principe. Maximum correntropy Kalman filter. *Automatica*, 76:70–77, 2017.

[114] Badong Chen, Lujuan Dang, Yuantao Gu, Nanning Zheng, and José C Príncipe. Minimum error entropy kalman filter. *IEEE Transaction on Systems, Man, and Cybernetics: Systems*, 51(9):5819–5829, 2019.

[115] Jiacheng He, Gang Wang, Huijun Yu, JunMing Liu, and Bei Peng. Generalized minimum error entropy kalman filter for non-gaussian noise. *ISA transactions*, 2022.

[116] Zhenbing Qiu and Lei Guo. Improved cubature Kalman filter for spacecraft attitude estimation. *IEEE Transaction Instrument on Measurement*, 70:1–13, 2020.

[117] Mingming Bai, Chengjiao Sun, and Yonggang Zhang. A robust generalized t distribution based kalman filter. *IEEE Transactions on Aerospace and Electronic Systems*, 2022.

[118] Ahmad Reza Heravi and Ghosheh Abed Hodtani. A new information theoretic relation between minimum error entropy and maximum correntropy. *IEEE Signal Processing Letters*, 25(7):921–925, 2018.

[119] Badong Chen, Xi Liu, Haiquan Zhao, and Jose C Principe. Maximum correntropy kalman filter. *Automatica*, 76:70–77, 2017.

[120] Xi Liu, Hua Qu, Jihong Zhao, and Badong Chen. Extended kalman filter under maximum correntropy criterion. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 1733–1737. IEEE, 2016.

[121] Zhiyu Zhang, Jinzhe Qiu, and Wentao Ma. Adaptive extended kalman filter with correntropy loss for robust power system state estimation. *Entropy*, 21(3):293, 2019.

[122] Shanmou Chen, Qiangqiang Zhang, Tao Zhang, Lingcong Zhang, Lina Peng, and Shiyuan Wang. Robust state estimation with maximum correntropy rotating geometric unscented kalman filter. *IEEE Transactions on Instrumentation and Measurement*, 71:1–14, 2021.

[123] Jingjing He, Changku Sun, Baoshang Zhang, and Peng Wang. Maximum correntropy square-root cubature Kalman filter for non-gaussian measurement noise. *IEEE access*, 8:70162–70170, 2020.

[124] Xi Liu, Hua Qu, Jihong Zhao, and Pengcheng Yue. Maximum correntropy square-root cubature Kalman filter with application to sins/gps integrated systems. *ISA transactions*, 80:195–202, 2018.

[125] Wutao Qin, Xiaogang Wang, and Naigang Cui. Maximum correntropy sparse gauss–hermite quadrature filter and its application in tracking ballistic missile. *IET Radar, Sonar and Navigation*, 11(9):1388–1396, 2017.

[126] Jiang Liu, Bai-gen Cai, and Jian Wang. Cooperative localization of connected vehicles: Integrating gnss with dsrc using a robust cubature Kalman filter. *IEEE Transactions on Intelligent Transportation Systems*, 18(8):2111–2125, 2016.

[127] Yu Liu, Hong Wang, and Chaohuan Hou. Ukf based nonlinear filtering using minimum entropy criterion. *IEEE Transactions on Signal Processing*, 61(20):4988–4999, 2013.

[128] Lujuan Dang, Badong Chen, Yulong Huang, Yonggang Zhang, and Haiquan Zhao. Cubature Kalman filter under minimum error entropy with fiducial points for ins/gps integration. *IEEE/CAA Journal of Automatica Sinica*, 9(3):450–465, 2021.

[129] Wanli Wang, K Tse Chi, and Shiyuan Wang. Generalized correntropy sparse Gauss–Hermite quadrature filter for epidemic tracking on complex networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(5):2770–2778, 2021.

[130] Jianbo Shao, Wu Chen, Ya Zhang, Fei Yu, and Jiachong Chang. Adaptive multi-kernel size based maximum correntropy cubature kalman filter for the robust state estimation. *IEEE Sensors Journal*, 2022.

[131] Yulong Huang, Yonggang Zhang, Ning Li, Syed Mohsen Naqvi, and Jonathon Chambers. A robust student's t based cubature filter. In *2016 19th International Conference on Information Fusion (FUSION)*, pages 9–16. IEEE, 2016.

[132] Arpan Chattopadhyay and Urbashi Mitra. Security against false data injection attack in cyber-physical systems. *IEEE Transactions on Control of Network Systems*, 2019.

[133] Dan Zhang, Qing-Guo Wang, Gang Feng, Yang Shi, and Athanasios V Vasilakos. A survey on attack detection, estimation and control of industrial cyber–physical systems. *ISA transactions*, 116:1–16, 2021.

[134] Shuai Liu, Guoliang Wei, Yan Song, and Yurong Liu. Extended Kalman filtering for stochastic nonlinear systems with randomly occurring cyber attacks. *Neurocomputing*, 207:708–716, 2016.

[135] Yan Liu and Guang-Hong Yang. Event-triggered distributed state estimation for cyber-physical systems under dos attacks. *IEEE Transactions on Cybernetics*, 52(5):3620–3631, 2022.

[136] Wei Ao, Yongduan Song, and Changyun Wen. Distributed secure state estimation and control for cpss under sensor attacks. *IEEE Transactions on Cybernetics*, 50(1):259–269, 2020.

[137] Simo Sarkka. On unscented kalman filtering for state estimation of continuous-time nonlinear systems. *IEEE Transaction on Automatic Control*, 52(9):1631–1641, 2007.

[138] Abhinoy Kumar Singh, Sumit Kumar, Nagendra Kumar, and Rahul Radhakrishnan. Bayesian approximation filtering with false data attack on network. *IEEE Transactions on Aerospace and Electronic Systems*, 58(2):976–988, 2021.

[139] Shanmou Chen, Qiangqiang Zhang, Dongyuan Lin, and Shiyuan Wang. A class of nonlinear kalman filters under a generalized measurement model with false data injection attacks. *IEEE Signal Processing Letters*, 29:1187–1191, 2022.

[140] A Willsky and J Lo. Estimation for rotational processes with one degree of freedom–part ii: Discrete-time processes. *IEEE Transaction on Automatic Control*, 20(1):22–30, 1975.

[141] Monir Azmani, Serge Reboul, Jean-Bernard Choquel, and Mohammed Benjelloun. A recursive fusion filter for angular data. In *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 882–887. IEEE, 2009.

[142] Georges Stienne, Serge Reboul, Monir Azmani, Jean-Bernard Choquel, and Mohammed Benjelloun. A multi-temporal multi-sensor circular fusion filter. *Information Fusion*, 18:86–100, 2014.

[143] Johannes Traa and Paris Smaragdis. A wrapped Kalman filter for azimuthal speaker tracking. *IEEE Signal Processing Letters*, 20(12):1257–1260, 2013.

[144] Gerhard Kurz, Igor Gilitschenski, and Uwe D Hanebeck. Recursive nonlinear filtering for angular data based on circular distributions. In *2013 American Control Conference*, pages 5439–5445. IEEE, 2013.

[145] Gerhard Kurz, Igor Gilitschenski, and Uwe D Hanebeck. Nonlinear measurement update for estimation of angular systems based on circular distributions. In *2014 American Control Conference*, pages 5694–5699. IEEE, 2014.

[146] Gerhard Kurz. *Directional estimation for robotic beating heart surgery*, volume 17. KIT Scientific Publishing, 2015.

[147] Gerhard Kurz, Igor Gilitschenski, and Uwe D Hanebeck. Recursive Bayesian filtering in circular state spaces. *IEEE Aerospace and Electronic Systems Magazine*, 31(3):70–87, 2016.

[148] Eloy Garcia and Panos J Antsaklis. Model-based event-triggered control for systems with quantization and time-varying network delays. *IEEE Transaction on Automatic Control*, 58(2):422–434, 2012.

[149] Bei Yan, Hanoch Lev-Ari, and Aleksandar M Stanković. Networked state estimation with delayed and irregularly spaced time-stamped observations. *IEEE Transactions on Control of Network Systems*, 5(3):888–900, 2017.

[150] Jhon A Isaza, Hector A Botero, and Hernan Alvarez. State estimation using non-uniform and delayed information: A review. *International Journal of Automation and Computing*, 15(2):125–141, 2018.

[151] Abhinoy Kumar Singh, Shovan Bhaumik, and Paresh Date. Quadrature filters for one-step randomly delayed measurements. *Applied Mathematics and Computation*, 40(19-20):8296–8308, 2016.

[152] Shuai Liu, Zidong Wang, Yun Chen, and Guoliang Wei. Protocol-based unscented Kalman filtering in the presence of stochastic uncertainties. *IEEE Transactions on Automatic Control*, 65(3):1303–1309, 2019.

[153] Wutao Qin, Xiaogang Wang, and Naigang Cui. Robust filtering with randomly delayed measurements and its application to ballistic target tracking in boost phase. *Transactions of the Institute of Measurement and Control*, 41(7):2077–2088, 2019.

[154] Peter Harremoës. Binomial and poisson distributions as maximum entropy distributions. *IEEE Transactions on Information Theory*, 47(5):2039–2041, 2001.

[155] JBV Reddy, Pradipta K Dash, R Samantaray, and Alexander K Moharana. Fast tracking of power quality disturbance signals using an optimized unscented filter. *IEEE Transactions on Instrumentation and Measurement*, 58(12):3943–3952, 2009.

[156] Maciej Niedźwiecki and Piotr Kaczmarek. Estimation and tracking of complex-valued quasi-periodically varying systems. *Automatica*, 41(9):1503–1516, 2005.

[157] Guoqing Wang, Ning Li, and Yonggang Zhang. Maximum correntropy unscented Kalman and information filters for non-gaussian measurement noise. *Journal of the Franklin Institute*, 354(18):8659–8677, 2017.

[158] Shanmou Chen, Qiangqiang Zhang, Tao Zhang, Lingcong Zhang, Lina Peng, and Shiyuan Wang. Robust state estimation with maximum correntropy rotating geometric unscented Kalman filter. *IEEE Transactions on Instrumentation and Measurement*, 71:1–14, 2021.

[159] Guddu Kumar, Sumanta Kumar Nanda, Alok Kumar Verma, Vimal Bhatia, and Abhinoy Kumar Singh. Nonlinear Gaussian filtering with network-induced delay in measurements. *IEEE Transactions on Aerospace and Electronic Systems*, pages 1–7, 2022.

[160] Yi-zheng Zhao. An improved unscented particle filter with global sampling strategy. *Journal of Computational Engineering*, 2014, 2014.

[161] Xingzi Qiang, Yanbo Zhu, and Rui Xue. Svrpf: An improved particle filter for a nonlinear/non-Gaussian environment. *IEEE Access*, 7:151638–151651, 2019.

[162] Fahira Afzal Maken, Fabio Ramos, and Lionel Ott. Stein particle filter for nonlinear, non-Gaussian state estimation. *IEEE Robotics and Automation Letters*, 7(2):5421–5428, 2022.

[163] Xinke Deng, Arsalan Mousavian, Yu Xiang, Fei Xia, Timothy Bretl, and Dieter Fox. Poserbpf: A rao–blackwellized particle filter for 6-d object pose tracking. *IEEE Transactions on Robotics*, 37(5):1328–1342, 2021.

[164] Mohamed Ahwiadi and Wilson Wang. An enhanced mutated particle filter technique for system state estimation and battery life prediction. *IEEE Transactions on Instrumentation and Measurement*, 68(3):923–935, 2018.

[165] Matthew A Nicely and B Earl Wells. Improved parallel resampling methods for particle filtering. *IEEE Access*, 7:47593–47604, 2019.

[166] Manoel BL Aquino, João PF Guimarães, Aluisio IR Fontes, Leandro LS Linhares, Joilson BA Rego, and Allan De M Martins. Circular correntropy: Definition and application in impulsive noise environments. *IEEE Access*, 2022.

[167] Kanti V Mardia and Jochen Voss. Some fundamental properties of a multivariate von mises distribution. *Communications in Statistics-Theory and Methods*, 43(6):1132–1144, 2014.

[168] Barry Simon. *Orthogonal polynomials on the unit circle*. American Mathematical Soc., 2005.

[169] Ruymán Cruz-Barroso, Pablo González-Vera, and Francisco Perdomo-Pío. Quadrature formulas associated with Rogers–Szegő polynomials. *Computers & Mathematics with Applications*, 57(2):308–323, 2009.

[170] Anahita Nodehi, Mousa Golalizadeh, Mehdi Maadooliat, and Claudio Agostinelli. Estimation of multivariate wrapped models for data in torus. *arXiv preprint arXiv:1811.06007*, 2018.

[171] Ruymán Cruz-Barroso, Pablo González-Vera, and Francisco Perdomo-Pío. An application of szegő polynomials to the computation of certain weighted integrals on the real line. *Numerical algorithms*, 52(3):273–293, 2009.

[172] A Bultheel, R Cruz-Barroso, P González-Vera, and F Perdomo-Pıo. On the computation of symmetric szego-type quadrature formulas. In *Proceedings of the MCalv6 (5) Conference, Zaragoza*, volume 33, pages 177–196, 2009.

[173] Adhemar Bultheel, Pablo Gonzalez-Vera, Erik Hendriksen, and Olav Njåstad. Computation of rational szegő–lobatto quadrature formulas. *Applied numerical mathematics*, 60(12):1251–1263, 2010.