Distributed Estimation and Quantization Algorithms for Wireless Sensor Networks

by

Sahar Movaghati

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Communications

Department of Electrical and Computer Engineering
University of Alberta

Dedicated to my beloved parents,
and
my beautiful homeland, Iran.

Abstract

Wireless sensor networks (WSNs) consist of small sensor devices with limited power and processing capability, which cooperate through wireless transmission, in order to fulfill a common task. These networks are currently employed on land, underground, and underwater, in a wide range of applications including environmental sensing, industrial and structural monitoring, medical care, and etc. However, there are still many impediments that hold back these networks from being pervasive, some of which are characteristics of WSNs, such as scarcity of energy and bandwidth resources and limited processing and storage capability of sensor nodes. Therefore, many challenges still need to be overcome before WSNs can be extensively employed.

In this study, we concentrate on developing algorithms that are useful for WSN estimation tasks. In designing these algorithms we consider the special constraints and characteristics of WSNs, i.e., distributed nature of the measurements and the processing resources, as well as the limited sensors' energy. We first investigate a general stochastic inference problem in a WSN. We design a non-parametric algorithm for tracking a random process using sensors' distributed and noisy measurements. Next we narrow down the problem to parameter estimation in WSNs, and design distributed quantizers to compress sensors' data while maintaining an accurate estimation of the unknown parameter.

The contributions of this thesis are as follows. In Chapter 3, we design an algorithm for the distributed inference problem in WSNs. We first use factor graphs to model the stochastic dependencies among the problem variables, and factorize a centralized estimation problem to a number of local dependencies. A message passing algorithm called the sum-product algorithm is then used on the factor graph to determine local computations and data exchange that must be performed by sensors in order to achieve the estimation goal. To tackle the nonlinearities in the problem, we combine the particle filtering and Monte-Carlo sampling in the sum-product algorithm and develop a distributed non-parametric solution for the general

nonlinear inference problems in WSNs. We apply our algorithm to the problem of target tracking using WSNs, and show that even with a few number of particles the algorithm can efficiently track the target.

In Chapter 4, we focus on the parameter estimation problem in a WSN under energy limitations. In such problems, each sensor sends a compressed version of its noisy observation of the same parameter to the fusion centre, where the parameter is estimated from the received data. In Chapter 4, we design a set of local quantizers that quantize each sensor's measurement to a few bits. We optimize the quantizers' design by maximizing the mutual information between the quantized data and the unknown parameter. At the fusion centre, we design the appropriate decoder that incorporates the compressed data from all sensors to estimate the parameter.

In Chapters 5 and 6, we narrow down the problem to very stringent capacity constraints, where each sensor quantizes its data to exactly one bit. In Chapter 5, we find a set of local binary quantizers that together act as a multi-level distributed quantization. In Chapter 6, we address inhomogeneous WSNs, where sensors have different signal-to-noise ratios. We devise an algorithm using the Hungarian method to find the best sensor-quantizer assignment that minimizes the estimation error.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my advisor Professor Masoud Ardakani for his continuous support in every step of my PhD, for being patient and supportive during the up and downs of this season of my life, for being always there to give ideas, guidance and motivation. Without him being a great teacher, my PhD experience would have never been a great stage of my life.

I would also like to express my appreciation to the rest of my thesis committee, Professor Chintha Tellambura and Professor Yindi Jing, for their insightful comments, and helpful ideas about my work, and Professor Soosan Beheshti and Professor Petr Musilek for reviewing and providing thoughtful comments about my thesis. I owe a debt of gratitude to the Department of Electrical and Computer Engineering and its staff, specifically Ms. Pinder Bains, for providing a smooth and relax environment for conducting my studies and research. I would also like to show my appreciation to the Faculty of Graduate Studies and Research at the University of Alberta, and the Alberta Innovates for their financial fundings towards this thesis.

I thank my friends for making my work time and life time more valuable and cheerful. I was lucky to have the company of my brilliant fellow lab-mates, Moslem, Mahdi, Raman, Zohreh, Gayan, and Damith, and all my friends in Edmonton who helped me not to feel away from home and family.

Last but not least, I would like to thank my family, especially my parents, Nasrin and Hadi, for always believing in me, and for their continuous love and support all through my life. Without them I would never be able to make a step in my life. I thank my younger brother and sister for adding joy and value to my life, and finally my fianc, Ali, not just for giving energy, support, and encouragement during final steps of my thesis but furthermore for bringing a new wonderful value to my life.

Table of Contents

List of Tables

List of Figures

List of Symbols

Symbol	Definition	First Use
$p(\cdot)$	Probability density function	7
$\mu_{x\to f}$	Message from a variable node to a function node	9
$\mu_{f\to x}$	Message from a function node to a variable node	9
$\mathcal{N}$	Number of particles in Monte Carlo integration	12
$E\{X\}$	Expected value of X	12
w^i	Importance weights	12
$q(x)$	Importance density	12
$\mathcal{F}$	Number of input messages to a variable node in a FG	12
$D(R)$	Distortion-rate function	15
I	Mutual information	15
$\mathcal{Q}(\cdot)$	Centralized quantizer	16
L	Number of quantization levels	16
$\sigma^2_{\hat{X}}$	MSE	16
I_F	Fisher information	18
$\Gamma_{b,n}$	Cost of Assigning Task b to Agent n	18
π	Bit-sensor assignment	18
$\mathcal{C}(\pi)$	Total Cost of Assignment π	18
$G(U,V;W)$	Bipartite graph	19
$\mathcal{M}$	A matching in a bipartite graph	19
$\ell(\cdot)$	Feasible labeling for the vertices of the graph	20
G_ℓ	Equality graph according to labeling ℓ	20
$\mathcal{V}$	Number of input messages to a function node in a FG	29

$\mathcal{S}_k$	Number of sensors tracking the target at time k	33
r	Association hypothesis variable	44
Y	Noisy measurements	52
W	Measurement noise	52
Z	Quantized measurements	52
T	Received symbol at channel output	52

List of Abbreviations

Abbreviation	Description	First Use
WSN	Wireless sensor network	1
FC	Fusion centre	2
FG	Factor graph	4
MPA	Message passing algorithm	4
SPA	Sum-product algorithm	4
PF	Particle filtering	4
MI	Mutual information	5
pdf	Probability density function	7
ML	Maximum likelihood	8
MMSE	Minimum mean squared error	8
KF	Kalman filter	8
MSE	Mean squared error	15
CRLB	Cramér-Rao lower bound	18
BFS	Breadth first search	22
DFS	Depth first search	22
PF-based MPA	Particle filter based message passing algorithm	27
SNR	signal-to-noise ratio	41
EKF	extended Kalman filter	41
BSC	Binary symmetric channel	50
LBQP	local binary quantization pattern	67
BEP	Bit error probability	77
MAP	Maximum a posteriori	79

Chapter 1

Introduction

1.1 Motivation

A wireless sensor network (WSN) is a collection of autonomous sensor nodes with limited power and processing capability, which are physically distributed in an area, cooperating with each other to collect and process data from the surroundings [1]. Depending on the application of the WSN, the sensor nodes can observe one or more physical properties, such as, temperature, sound, motion, electromagnetic radiation, pressure, etc. WSNs have many applications such as environmental monitoring (e.g., air, water, soil, micro-climate, animals, etc.), medical care and health monitoring, smart home and office, agriculture and irrigation management, intelligent transportation and traffic monitoring, air quality and air pollution monitoring, fire detection, object localization and tracking, intruder detecting, area surveillance, etc. [1].

For performing wireless transmission and reception, sensor nodes in a WSN are equipped with wireless communication devices, i.e. radio transceivers. Depending on the application, the transceivers could implement different protocols, including ZigBee, 802.15.4, WiFi and RF, ranging from megahertz to gigahertz in frequency spectrum [2]. The nodes usually have some capability of data processing and computations. The energy source of the sensor is in the form of a battery with limited power. Also, there is some memory unit in a sensor node for restoring the observations and the processed data. Rapid progress in recent technologies including Micro-electromechanical systems (MEMS) has enabled industrialization of low-cost and smaller sensor devices. Today, the industrial sensor devices are a few inches in dimension, including their battery and wireless transceivers. Furthermore, the advances in communication technologies and network protocols pledge

promising future for the large deployment of sensor networks, including the Internet-of-Things [3]. It is predicted that there will be 25 billion devices connected to the internet by 2015 and 50 billion by 2020 [4].

Despite all the progress, there are still many impediments which introduce challenges in the pervasive application of WSNs. Among them some important challenges include, limited power and bandwidth resources, limited processing and storage capability of sensor nodes, and the random and distributed nature of these networks. Due to these unique characteristics of WSNs, there are more challenges for designing WSN algorithms and protocols compared to conventional wired networks or other fixed or mobile wireless networks.

Limited energy recourses is one of the most crucial limitations that confront WSN applications. The energy resources are mainly consumed for the processing tasks and wireless transmission. Therefore, it is necessary to design algorithms that manage the data processing and data transmissions in the WSN in a way to preserve network resources for long-term monitoring, while achieving a good quality for the estimation tasks. In this thesis we have addressed the design of such algorithms.

To design appropriate algorithms for WSNs, we have to overcome an extra challenge. That is, the local observed data are by nature physically distributed over the network, and are not available in one processing center. Therefore, many global algorithms such as routing methods, data management methods, tracking and detection algorithms, must be designed as distributed algorithms for WSNs. In a distributed algorithm, each sensor performs some local processing as part of the global task (e.g., estimation or tracking), and there is less dependence on a central node or fusion centre (FC). In this thesis we devise distributed solutions for inference problems in a WSN, while maintaining the desired accuracy in estimation. These challenges are further explained in the following two sections.

1.2 Distributed Inference

In Chapter 3, we focus on the use of WSNs for distributed inference and design some algorithms for this purpose. WSNs are distributed sensing systems because they consist of a set of physically distributed measurement devices with wireless communication capability. As a distributed sensing system they can be employed as a set of humidity sensors that take measurements for an irrigation management system, an antenna array that is deployed to locate and track a moving vehicle, the

wearable biomedical signal measurement devices that are used for health monitoring, etc. In all these applications the sensors take some measurement of a physical property, i.e., temperature, humidity, distance to an object, heart bit rate, etc. The sensors' measurement data must be processed in the sensors and/or in the FC to achieve the common goal of the system, e.g., optimizing the plants' watering schedule, tracking a moving object trajectory, or assuring the health condition of the patient. The signal processing algorithms of WSNs control the way sensors' data are processed locally, how the sensors' data are aggregated and communicated to other nodes, and how these data are combined in a FC.

However, in many examples of WSNs, there is no anchor node or FC, which can be reached by other sensors and be relied on for central processing tasks. In such ad-hoc networks, the global task must be performed in a distributed fashion, i.e., sensors must cooperatively perform all the processing tasks by doing local processing and sharing data among each other through local communication links. In such WSNs a *distributed signal processing algorithm* determines how to locally process the sensors' measurements at the sensors, what to send to the other sensors, and how to combine the received data from neighboring sensors, in order to achieve the global task of the system.

There have been many studies for developing distributed signal processing algorithms, such as, distributed consensus [5–7], distributed localization and tracking [8–10], distributed link loss monitoring [11, 12], distributed compression and quantization [13, 14], distributed estimation [15–17], distributed detection [18], etc. In all these algorithms the overall task is broken down to local processing tasks and sensor collaboration.

In this study, we develop a distributed signal processing algorithm for WSNs to enable them perform a general category of inference and estimation tasks. We assume a model of a WSN consisting of a number of distributed sensors that aim to collaboratively estimate the state of a random object in time, e.g., the trajectory of a moving abject, the water level of a river, the production rate in oil wells, etc. In such applications, sensors have local measurements of the object (i.e., a random process) and need to cooperatively process these measurements to infer the state of the object in time. Our distributed inference algorithm, achieves the global estimation task by determining how each sensor processes its local partial measurement and the partial measurements it receives form its neighboring sensors.

The majority of the advanced tasks performed by WSNs, such as data fusion, decision making, hypothesis testing, detection, etc., can be interpreted in the form of a stochastic inference problem. In a stochastic inference problem, the phenomenon of interest, i.e., target trajectory, water level of a river, production rate of a well, heart bit rate of a patient, etc., are modeled as a random process. Each senor, making some (generally noisy) measurements from some attributes related to this random process, have some partial (and usually distorted) information about the process. In this setup, a stochastic inference problem is about inferring the unknown random process from the local distorted measurements of the sensors.

In our distributed algorithm for stochastic inference in WSNs, we benefit from the factor graph (FG) modeling and a message passing algorithm (MPA), called the sum-product algorithm (SPA) to factorize a centralized inference problem into local computations plus some information passing among neighboring sensors. We combine the particle filtering (PF) and Monte-Carlo sampling methods in the message passing algorithm to develop a distributed non-parametric solution for the general nonlinear inference problems in WSNs.

1.3 Distributed Quantization

In many WSN applications the FC has to estimate an unknown random parameter based on some noisy observations received from the sensors. In these applications, sensors' measurement data need to be transmitted to the FC over the wireless links. This wireless transmission is usually the most power-consuming task in the sensors [19]. Since sensors' power resources are limited, the tasks that involve frequent communication between the sensors and the FC face a challenge. In fact, as the sensors' size shrink, the amount of energy they can store also reduces. Batteries are normally the power source within a sensor for all its tasks, hence sensors performance and lifetime is restrained to their battery. Therefore, research is ongoing for utilizing the sensors' energy efficiently without sacrificing the performance [20].

Due to these resource constraints, the distributed signal processing algorithms need to be energy-efficient in a WSN. One way to achieve this efficiency is to compress the sensors' local observations before transmitting to the FC [21]. As a solution, many studies are focused on reducing the number of transmitted bits from the sensors to the FC via quantization. One way is to use the same approach as in the centralized quantization. The optimal solution for the centralized quantization is

given by the Lloyd-Max algorithm [22, 23]. It finds the optimum centralized quantizer using an iterative algorithm that decreases the quantization distortion at each iteration.

In a WSN, the centralized quantization techniques of [22–25] can be used for designing a local quantizer for every sensor to compress its data before transmission. However, since sensors' measurement data are usually correlated (for example when sensors measure the same parameter), optimum estimation performance can be obtained through joint design of the local quantizers. Therefore, [26–29] jointly design N local quantizers using cyclic algorithms based on alternating minimization [30]. These algorithms are computationally complex. Thus, less complex but not optimal approaches have been studied in [13–17, 31–38].

In Chapter 4, we jointly design N quantizers by using the mutual information (MI) as the optimization measure instead of the estimation error as in [22–25]. We design quantizers that maximize the MI of the quantized data and the unknown parameter. Our motivation for using the MI is that it is a fundamental measure showing how much information the quantized variables contain about the unknown variable. Hence, we design a set of quantizers that quantize the noisy measurements into a set of variables that together contain the most information about the unknown parameter. MI allows to incorporate the effect of communication channels in the design of quantizers. Therefore, we find optimal channel-aware local quantizers by integrating the channel information in our design.

For very stringent capacity constraints in WSNs, each sensor is only allowed to send one bit per measurement. In those cases we offer a design method that chooses the quantization edges among a predetermined limited set of edges. Unlike iterative approaches that usually find local optimum solutions we use brute-force search with heuristic simplifications to find a global solution. Once the N quantization rules are obtained, one needs to assign each quantization rule to a sensor. This assignment of sensors and quantization rules can affect the estimation performance in inhomogeneous environments, where sensors measurement qualities are not the same. Therefore, in Chapter 6, we propose an algorithm based on Hungarian method to efficiently optimize the assignments for achieving the minimum estimation error.

1.4 Thesis Organization

This thesis is organized as follows: The theoretical preliminaries of all the topics covered in this thesis are explained in Chapter 2. The contributions are organized in four chapters. In Chapter 3, we develop an algorithm based on factor graph modeling and a message-passing algorithm for distributed estimation and inference in WSNs. In Chapter 4 we focus on the design of distributed quantization scheme. The optimum quantization rules are found by an iterative algorithm which maximizes the mutual information. Later in this chapter, we improve our design algorithm to integrate the channel error and achieve channel-aware distributed quantization scheme. In Chapter 5 we narrow down the problem to binary quantization per sensor and find the best N binary quantization rules. In Chapter 6, we find the optimum assignment between the local quantizers and the sensor nodes, for inhomogeneous environments. Finally, in Chapter 7, we conclude and discuss our contributions and suggest some future research topics.

Chapter 2

Background

2.1 Inference Problems

2.1.1 Setup

In recent decades stochastic approaches, such as Bayesian methods [39], have become popular for solving many problems. In stochastic inference problems, the phenomenon of interest as well as the sensors measurement data are represented by random variables or processes. The resulting model is called a stochastic model for the inference problem.

In a stochastic approach for solving the problems, all quantities involved in the problem, such as the measurements (observed variables) and the unknowns (hidden or unobserved variables) are treated as random variables. The uncertain/ambiguous relations among the variables of the problem are modeled via stochastic relations among the random variables. For example, assume that the random variables $y_1, y_2, \cdots, y_N$ represent N measurement quantities and $x_1, x_2, \cdots, x_M$ stand for M unknowns. The following implications are considered in stochastic approaches.

i) The knowledge or belief associated with each random variable is described by a probability distribution function (pdf). For example, the pdf $p(x_m)$ explains some a priori knowledge about the unknown x_m. The more the variance of this pdf, the more is the uncertainty about the value of x_m.

ii) A joint probability distribution of the variables captures the stochastic relationship among the variables involved in the problem. For example,

$$p(x_1, x_2, \cdots, x_M, y_1, y_2, \cdots, y_N). \tag{2.1}$$

iii) The a posteriori belief about the set of variables $x_1, x_2, \cdots, x_M$ is a condi-

tional pdf like

$$p(x_1, x_2, \cdots, x_M | y_1, y_2, \cdots, y_N), \tag{2.2}$$

which encapsulates all the information that could be achieved about the unknowns $x_1, x_2, \cdots, x_M$ knowing the measurements $y_1, y_2, \cdots, y_N$ [40, 41]. Normally, the amount of uncertainty in the a posteriori distribution is less than the uncertainty in the a priori distribution.

The goal in a stochastic inference is to find the marginals of the a posteriori distributions. For example, the marginal a posteriori distribution of x_m, i.e.,

$$\begin{aligned} &p(x_m | y_1, y_2, \cdots, y_N) \\ &= \int p(x_1, x_2, \cdots, x_M | y_1, y_2, \cdots, y_N) dx_1 \cdots dx_{m-1} dx_{m+1} dx_M, \end{aligned} \tag{2.3}$$

is the a posteriori belief about the unknown x_m. If the pdf (2.3) is known, any desired estimate of x_m can be made regarding an estimation metric, for example, a maximum likelihood (ML) or a minimum mean squared error (MMSE) estimation.

The above described stochastic setup can be applied to many problems ranging from single, time-invariant parameter estimation to multi-variant time-varying processes. To solve these problems, i.e., to find the a posteriori marginal pdfs (2.3), many techniques have been developed, ranging from the Bayesian hypothesis testing to the Kalman filter [42], PF [43–45], and graph-based methods such as belief propagation [46] and the SPA [47]. In Section 2.1.2, we explain the graphical modeling approach for solving inference problems.

2.1.2 Graphical Models

A graphical model is an appropriate tool that can be used to build a solution for a general stochastic inference problem. Graphical models such as FGs have been used to model stochastic relationships of the variables involved in various inference problem. The FG takes advantage of the conditional independencies among the variables to break down a global multi-variate stochastic relation like the pdf (5.11) to smaller local functions. Therefore, a complicated global processing on the entire set of variables can be broken down to smaller processing modules that involve only some of the variables. Specifically, instead of calculating multi-variable integrals to find the marginal beliefs (2.3), the FG representation of the stochastic problem helps derive a set of simpler processing rules that together achieve the same performance

as the centralized solution. To understand how FGs help in inference problems, the concept of the FG and the message passing algorithm is briefly described here.

Assume a global function of a set of variables $\{x_1, x_2, \cdots, x_n\}$, i.e., $\zeta(x_1, x_2, \cdots, x_n)$, factors into a product of several local functions, each having a subset of $\{x_1, x_2, \cdots, x_n\}$ as their arguments. In other words,

$$\zeta(x_1, x_2, \cdots, x_n) = \prod_j f_j(X_j), \tag{2.4}$$

where X_j is a subset of $\{x_1, x_2, \cdots, x_n\}$, and $f_j(X_j)$ is a function having the elements of X_j as its arguments. The factorization in (2.4) can be shown using a bipartite graph called factor graph, which has a *variable node* for each variable x_i, and a *function node* for each local function f_j. There is an edge between the variable node x_i and the function node f_j if and only if x_i is an argument of f_j [47]. For example, if $\zeta(x_1, x_2, x_3, x_4, x_5)$ can be expressed as

$$\zeta(x_1, x_2, x_3, x_4, x_5) = f_1(x_1, x_2) f_2(x_3) f_3(x_2, x_3, x_4) f_4(x_4, x_5), \tag{2.5}$$

the FG corresponding to this factorization is shown is Fig. 2.1. In order to find all marginal functions of a global function some calculations can be done on the FG according to a MPA. A MPA is usually described by defining the *messages* that are passed along the edges of the FG and the *update rules* associated with the messages as they move through the nodes of the graph. A message sent along an edge is a function of the connected variable node to that edge and usually carries information about that variable. The message update rules are procedures performed in the graph nodes to manipulate the incoming messages to the node in order to generate output messages from the node. A very famous MPA called the SPA is introduced in [47]. By running the SPA on a cycle-free FG, once a message is passed in both directions on every edge, the marginal function of any variable can be found.

The SPA establishes the message update rules at the variable node x and the function node f. We represent the message going from a variable node to a function node as $\mu_{x\to f}(x)$ and a message from a function node to a variable node as $\mu_{f\to x}(x)$. Note that both messages are functions of the particular variable node involved in the transaction. Let $\aleph(x)$ indicate the set of neighbors of the variable node x and $\aleph(f)$ indicate the set of neighbors for the function node f. Then the update rules of the SPA are formulated as

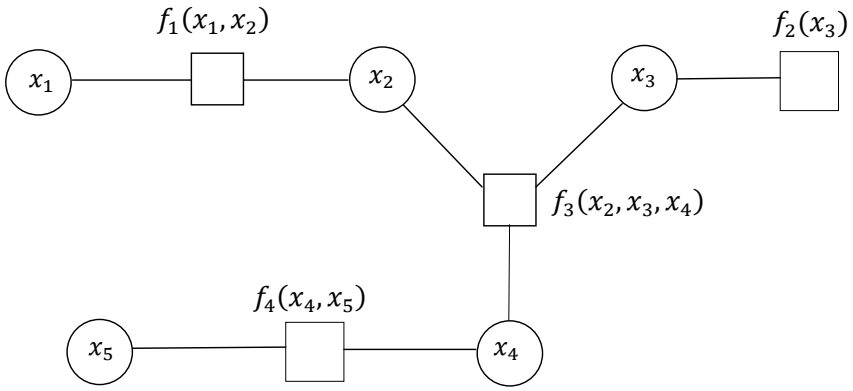

Figure 2.1: Factor graph representation for the function ζ in (2.5).

Update Rule 1:

$$\mu_{x\to f}(x) = \prod_{\lambda\in\aleph(x)\setminus f} \mu_{\lambda\to x}(x) \tag{2.6}$$

Update Rule 2:

$$\mu_{f\to x}(x) = \sum_{\sim x}(f(X) \prod_{y\in\aleph(f)\setminus x} \mu_{y\to f}(y)) \tag{2.7}$$

where X is the set of the arguments of the local function f which will be the same as $\aleph(f)$. The notation $\aleph(x) \setminus f$ denotes the set of all the neighbors of x excluding the function node f. Also, $\sim x$ means that the sum (or the integral in the case of continuous variable) is taken on all the variables except x.

The applications of FG and the SPA are very extensive. Signal processing, digital communication and artificial intelligence algorithms can be derived as instances of the SPA. In the Bayesian estimation problems specifically, the goal is to infer some information or "belief" about some hidden (unobserved) variables $x_1, x_2, \cdots, x_M$ based on some observed variables $y_1, y_2, \cdots, y_N$. In other words, we are looking for marginal distributions of a pdf function like $p(x_1, x_2, \cdots, x_M|y_1, y_2, \cdots, y_N)$. We can use the inter-dependencies of the variables to factorize this function into smaller local functions similar to the scheme in (2.4). Once we do that, we can associate a FG with the factorization and run the SPA update rules on the graph to find the marginal pdfs of the unknown variables. The SPA determines the mechanism by which local messages are exchanged between the neighboring nodes on the graph. For inference problems in WSNs, this approach can be very beneficial, because it breaks down the centralized calculations into smaller local update rules.

The message update rules (2.6) and (2.7) specify a set of analytical calculations which could be very complex in general. Therefore, The application of the SPA for inference problems is often restricted to problems involving binary or discrete variables or continuous variables with Gaussian distributions. In some applications, such as coding or behavioral modeling [47], the variables are discrete and belong to a limited alphabet. In those cases, the messages reduce to a numerical table instead of continuous functions and the update rules become numerical calculations. However, in a general continuous stochastic inference problem, the messages are usually continuous functions which are pdfs or conditional pdfs. Working with arbitrary pdfs and trying to solve (2.6) and (2.7) analytically is usually impossible or very complicated. Some approaches to deal with the infeasible equations of message update rules are suggested in [48]. For those inference problems that involve only Gaussian posteriori distributions (linear and Gaussian models), the message update rules of MPA reduce to simple calculations on mean and variance of the distributions. For details of these calculations, see [47,49,50]. For general stochastic problems with continuous pdfs, the integrals in rule 2 can be solved with the help of Monte-Carlo integration. This is explained in Section 2.1.3. Also, the multiplication of the pdfs can be done using a Gibbs sampler explained in Section 2.1.4.

2.1.3 Monte Carlo Integration

For non-Gaussian and non-linear inference problems, Monte-Carlo based solutions have been conventionally used in the literature. Monte Carlo integration [43] is a probabilistic method used to approximate an integral. In contrast to the Riemann sum method which blindly divides the variable space, in Monte Carlo integration a set of randomly chosen samples, called particles, represent the distribution of a random variable in its domain. Owing to the wise selection of the particles, Monte Carlo methods can be more accurate than Riemann sum approaches for the same computational complexity. This difference is more pronounced when the dimension of the integral is higher. In fact, an important property of Monte Carlo integration is that its rate of convergence is independent of the dimension of the integral [50].

Suppose that we want to numerically evaluate the integral [43]

$$\mathbb{I} = \int r(x)dx, \tag{2.8}$$

where $x \in \mathbb{R}^n$, $n \geq 0$. Monte Carlo methods for numerical integration first factorize

the integrand as $r(x) = s(x)\pi(x)$, so that $\pi(x)$ is a pdf. In other words it satisfies the conditions $\pi(x) \geq 0$ and $\int \pi(x)dx = 1$. Assume that it is practically possible to draw $\mathcal{N} \gg 1$ samples from the distribution π as x^i; $i = 1, \cdots, \mathcal{N}$. Therefore, the Monte Carlo estimate of the integral $\mathbb{I}$ is the sample mean

$$\mathbb{I} = \int s(x)\pi(x)dx = E\{s(x)\} \simeq \mathbb{I}_{\mathcal{N}} = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} s(x^i). \tag{2.9}$$

If the samples are independent $\mathbb{I}_{\mathcal{N}}$ is an unbiased estimation of $\mathbb{I}$, converging to $\mathbb{I}$ as $\mathcal{N} \to \infty$. Ideally, we want to generate samples from $\pi(x)$ and use (2.9) to estimate $\mathbb{I}$. However, suppose that we can only generate samples from another pdf $q(x)$. Then, we can still estimate $\mathbb{I}$ by applying some proper sample weighting, if the support of $\pi(x)$ and $q(x)$ are the same, i.e., $\pi(x) > 0$ implies $q(x) > 0$ for all x. Then, any integral of the form (2.8) can be written as

$$\mathbb{I} = \int s(x)\pi(x)dx = \int s(x)\frac{\pi(x)}{q(x)}q(x)dx. \tag{2.10}$$

Generating independent samples from $q(x)$ as $\{x^i; i = 1, \cdots, \mathcal{N}\}$, $\mathbb{I}$ can be approximated by the weighted sum

$$\mathbb{I}_{\mathcal{N}} = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} s(x^i)w^i, \qquad \text{where} \quad w^i = \frac{\pi(x^i)}{q(x^i)}. \tag{2.11}$$

w^i are called the importance weights, $q(x)$ is called the importance density, and the method is called importance sampling [43].

2.1.4 Gibbs Sampler

At a variable node x of a FG, the incoming messages must be multiplied to form the outgoing message. Assume that in a general structure, $\mathcal{F} + 1$ function nodes, $f_1, \ldots, f_{\mathcal{F}+1}$, are connected to a variable node x, as shown in Fig. 2.2. We now explain an algorithm using Gibbs sampler [51] to find the outgoing message from x to $f_{\mathcal{F}+1}$.

Suppose that each of the $\mathcal{F}$ incoming messages toward x are presented by a set of $\mathcal{N}$ particles and their importance weights, i.e. $\mu_{f_l \to x}(x) = h_l(x) = \{x_l^i, w_l^i; 1 \leq i \leq \mathcal{N}\}; 1 \leq l \leq \mathcal{F}$. We represent each incoming message $h_l(x)$ with a Gaussian mixture of $\mathcal{N}$ weighted Gaussian kernels. The ith Gaussian kernel of the lth message has its mean on x_l^i and has a weight equal to w_l^i. For simplicity we choose all the

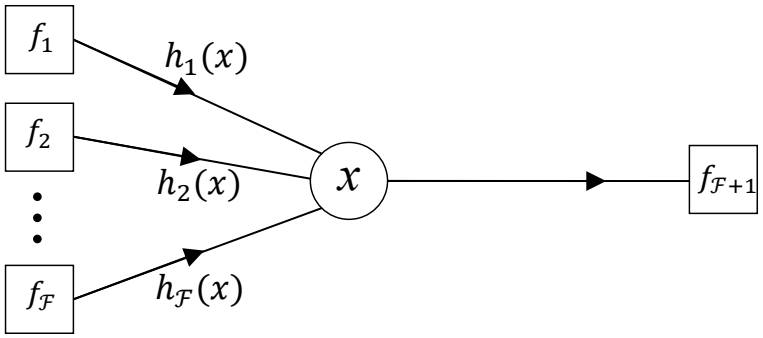

Figure 2.2: A variable node in a factor graph.

kernel variances of the lth Gaussian mixture equal to σ_l^2 [1]. A rule of thumb is to set σ_l^2 equal to the weighted variance of the samples of the lth message divided by $\mathcal{N}^{1/6}$ [52]. Hence, each incoming message is

$$\mu_{f_l \to x}(x) = h_l(x) = \sum_{i=1}^{\mathcal{N}} w_l^i \mathcal{N}(x; x_l^i, \sigma_l^2) \tag{2.12}$$

where $\mathcal{N}(x; x^i, \sigma_l^2)$ is a Gaussian function of the variable x, with mean x_l^i and variance σ_l^2. According to (2.6), the outgoing message $\mu_{x \to f_{\mathcal{F}+1}}(x)$ will be written as

$$\mu_{x \to f_{\mathcal{F}+1}}(x) = g(x) = \prod_{l=1}^{\mathcal{F}} \sum_{i=1}^{\mathcal{N}} w_l^i \mathcal{N}(x; x_l^i, \sigma_l^2) \tag{2.13}$$

Equation (2.13) is a product of $\mathcal{F}$ Gaussian mixtures, each with $\mathcal{N}$ components. Since the product of any number of Gaussian functions is proportional to a Gaussian function, by expanding (2.13), we have $\mathcal{N}^{\mathcal{F}}$ Gaussian components. Each component has a new mean, variance and weight. Our goal is to find $\mu_{x \to f_{\mathcal{F}+1}}(x) = g(x)$ as a set of $\mathcal{N}$ weighted particles to send to the next node on the graph $f_{\mathcal{F}+1}$. More specifically, we have to find a way to draw $\mathcal{N}$ samples from (2.13). Obviously, working with $\mathcal{N}^{\mathcal{F}}$ components is not an efficient solution. Thus, we use the Gibbs sampling method to draw the samples of $g(x)$. The Gibbs Sampler has been previously used in Bayesian networks to find the product of Gaussian mixtures [51]. It provides a way to draw asymptotically unbiased samples from a product of $\mathcal{F}$ Gaussian mixtures, each having $\mathcal{N}$ components.

The details of Gibbs sampler for message update at a variable node is given in Algorithm 1. For further details about the Gibbs sampler, please see [51].

[1] If a random variable is not one dimensional, the variance is replaced by the covariance matrix. The procedure remains the same.

Algorithm 1 Gibbs sampler for message update at a variable node.

Let the lth incoming messages, $1 \leq l \leq \mathcal{F}$, be represented by a Gaussian mixtures of $\mathcal{N}$ components, $\{w_l^i, x_l^i, \sigma_l^2\}_{i=1}^{\mathcal{N}}$

Construct the initial labels set $B^0 = \{b_1, ..., b_{\mathcal{F}}\}$ by sampling b_l according to the lth mixture weights, $p(b_l = i) \propto w_l^i$

repeat

for $l = 1$ to $\mathcal{F}$ **do**

Find the equivalent mean $\bar{m}$ and variance $\bar{\varsigma}^2$ of the product,

$$\prod_{j \neq l} \mathcal{N}(x; x_j^{b_j}, \sigma_j^2)$$

according to

$$\frac{1}{\bar{\varsigma}^2} = \sum_{j \neq l} \frac{1}{\sigma_j^2} \qquad \bar{m} = \bar{\varsigma}^2 \sum_{j \neq l} \frac{x_j^{b_j}}{\sigma_j^2}$$

for $i = 1$ to $\mathcal{N}$ **do**

Find the equivalent mean $\tilde{m}_i$ and variance $\tilde{\varsigma}_i^2$ of product $\mathcal{N}(x; \bar{m}, \bar{\varsigma}^2) \cdot \mathcal{N}(x; x_l^i, \sigma_l^2)$

$$\frac{1}{\tilde{\varsigma}_i^2} = \frac{1}{\bar{\varsigma}^2} + \frac{1}{\sigma_l^2} \qquad \tilde{m}_i = \tilde{\varsigma}_i^2 \left(\frac{\bar{m}}{\bar{\varsigma}^2} + \frac{x_l^i}{\sigma_l^2} \right)$$

From the above, find the conditional weight for the ith component of the lth mixture as

$$\tilde{w}_l^i = w_l^i \frac{\mathcal{N}(x; \bar{m}, \bar{\varsigma}^2) \cdot \mathcal{N}(x; x_l^i, \sigma_l^2)}{\mathcal{N}(x; \tilde{m}_i, \tilde{\varsigma}_i^2)}$$

end for

Draw a new label for b_l according to the weights $\{\tilde{w}_l^i\}_{i=1}^{\mathcal{N}}$ and update the labels set by replacing b_l instead of the previous label.

end for

until t iterations

Calculate the mean and variance of the equivalent Gaussian from the final set of labels and draw a sample from this Gaussian distribution.

2.2 Quantization and Estimation Theory

2.2.1 Quantization Theory and Rate-Distortion

A random source which has continuous amplitude requires infinite number of bits to be described. However, due to practical constraints, e.g., limited storage or channel capacity, only finite number of bits can be accommodated. To compress a continuous-amplitude source into limited amount of information (bits) its amplitude x should be quantized to $Q(x)$ which takes values from a discrete finite set. The average number of bits that is used to represent a quantized sample of X is called the quantization rate R. A side effect of quantization is some loss of information about X which depends on the quantization rate. This loss of information is defined in terms of a distortion measure [53]. The rate-distortion theory, first introduced by Shannon [54] and elaborated in [53], describes the relation between the amount of distortion caused by the quantization, and the rate by which the quantized source can be presented. Particularly, for a continuous-amplitude source with a specific known pdf, the rate-distortion theory gives a lower bound on the amount of distortion caused by a special quantization rate.

Assume that $\hat{X} = Q(X)$ is the quantized representation of X. First a distortion function $d(x, \hat{x}) \rightarrow \mathbb{R}^+$ must be defined as a cost of representing X with $\hat{X}$. A common distortion function is the squared-error distortion, i.e., $d(x, \hat{x}) = (x - \hat{x})^2$. The distortion is the expected value of the distortion function, i.e., $D = E\{(X - \hat{X})^2\}$, which is also called the mean squared error (MSE) . For a random source, the distortion-rate function $D(R)$ is the infimum of distortions D that is achievable for a given rate R [53]. The distortion-rate function for a random source X with pdf $p(x)$ is given by

$$D(R) = \min_{\substack{p(\hat{X}|X) \\ I(X;\hat{X}) \leq R}} E\{(X - \hat{X})^2\}, \tag{2.14}$$

where $I(X; \hat{X})$ is the mutual information of X and $\hat{X}$.

The distortion-rate function $D(R)$, determines the minimum distortion that can be achieved when an analog source is compressed with a rate R. The distortion-rate function for centralized estimation of a Gaussian source from a noisy observation under a rate constraint R is derived in [55], and is used as a lower bound for the distortion in the distributed estimation-quantization problem. For a scalar parameter with Gaussian distribution $\mathcal{N}(0, \sigma_X^2)$, which is observed through additive Gaussian

noise with variance σ^2, $D(R)$ is calculated as [55]

$$D(R) = \sigma_X^2 - \frac{\sigma_X^4}{\sigma_X^2 + \sigma^2}(1 - 2^{-2R}). \tag{2.15}$$

For distributed quantization as it appears in WSN applications, the rate distortion theory should be modified to apply for such scenarios. In Chapter 4, we find a distortion rate function for the distributed quantization problem.

2.2.2 Lloyd-Max Algorithm

The theoretical limits described by the rate distortion function can only be asymptotically achieved by optimal source coding [53], i.e., arbitrary long sequences of symbols and random coding. On the other hand, representing a continuous-amplitude source by a limited set of symbols is a challenge itself. For a random scalar, the optimal quantization rule can be obtained based on a method introduced by Lloyd and Max [22,23]. They propose an iterative algorithm for finding the best (local optimum) quantization rule for a random source to achieve the lowest distortion. The distortion measure they use is the MSE. Using the Lloyd-Max algorithm, the optimal[2] L-level quantization rule which minimizes the estimation MSE can be found for a random scalar X with pdf $p(x)$. The joint quantizer design for multiple variables has been studied under vector quantization [56, 57].

Here, we describe the Lloyd-Max algorithm for the optimum L-level quantizer of a random real-valued scalar X distributed according to pdf $p(x)$. The quantizer $\mathcal{Q}(\cdot)$ assigns a value from the set of reconstruction points (or quanta or centroids) $q_1, q_2, \cdots, q_L$ to each $x \in \mathbb{R}$. In that manner, the quantizer partitions $\mathbb{R}$ to L disjoint subsets $\{S_1, S_2, \cdots, S_L\}$, associating each partition with a reconstruction point. The quantization design problem is to choose the partitions $\{S_1, S_2, \cdots, S_L\}$ and reconstruction points $q_1, q_2, \cdots, q_L$ that minimize the quantization noise (MSE), defined as $\sigma_{\hat{X}}^2$ as follows

$$\sigma_{\hat{X}}^2 = E\{(X - Q(X))^2\} = \sum_{i=1}^{L} \int_{S_i} (x - q_i)^2 p(x) dx. \tag{2.16}$$

By differentiating with respect to the unknown parameters, Lloyd and Max show that minimizing (2.16) requires satisfying the following two conditions.

[2] It must be mentioned that all iterative algorithms for quantization design find a local optimal solution which depends on the initial quantization rules.

(i), Given a set of reconstruction points $q_1, q_2, \cdots, q_L$, the MSE is minimized by choosing $\{S_1, S_2, \cdots, S_L\}$ to be the Voronoi partitions associated with the reconstruction points. This means each value of $X \in \mathbb{R}$ must be mapped to the reconstruction point that is closest to it. This means

$$S_i = \{x : |x - q_i| \leq |x - q_j|\}, \qquad 1 \leq j \leq L, j \neq i. \tag{2.17}$$

(ii), Given a set of partitions $\{S_1, S_2, \cdots, S_L\}$, the reconstruction points must be the center of mass of the corresponding partitions. That is,

$$q_i = \frac{\int_{S_i} xp(x)dx}{\int_{S_i} p(x)dx}. \tag{2.18}$$

Unfortunately, it is not easy to find a closed form solution that can satisfy both above conditions. However, a trial-and-error or iterative approach can be used to find the quantization rule. The algorithm can start by an initial guess for the partitions $S_1^{(0)}, S_2^{(0)}, \cdots, S_L^{(0)}$. Then, based on (2.18) the construction points $q_1^{(0)}, q_2^{(0)}, \cdots, q_L^{(0)}$ are found which result in an MSE of $\sigma_{\hat{X}}^{2\,(0)}$. The partitions are updated using (2.17) to find $S_1^{(1)}, S_2^{(1)}, \cdots, S_L^{(1)}$ and subsequently, the reconstruction points are updated again. This procedure continues at successive iterations by alternatively imposing conditions (2.18) and (2.17). The result of these successive trials is such that [22]

$$\sigma_{\hat{X}}^{2\,(0)} \geq \sigma_{\hat{X}}^{2\,(1)} \geq \sigma_{\hat{X}}^{2\,(2)} \cdots \tag{2.19}$$

Finally, $\sigma_{\hat{X}}^{2\,(\infty)}$ will converge to a local minimum of the MSE [22], and the final partitions and reconstruction points will decide the quantizer.

In this thesis, we deal with the optimum quantizer design problem. However, it defers from the basic Lloyd-Max problem in two ways. First, the quantizer is not performing on the samples of the random parameter X, but on the noisy version of the parameter, i.e., measurement Y. Another twist of the problem is the need to design quantizers for a set of N noisy measurements of the unknown parameter, i.e., $Y_1, Y_2, \cdots, Y_N$. In Chapter 4, we elaborate this problem and propose an iterative algorithm to solve the quantizer design problem for distributed noisy measurements of a random parameter.

2.2.3 Cramer-Rao Lower Bound

A basic problem in estimation is to determine a real (nonrandom) parameter X using an estimator $\hat{X}$ from a random variable Y. A better estimator is the one that has

lower estimation variance $\sigma^2_{\hat{X}}$. The greatest lower bound of the variance of unbiased estimators is determined by the Cramér-Rao lower bound (CRLB) [58]. The CRLB can be used to investigate whether a particular estimator is the optimum estimator or how close is to the optimum estimator. Assuming that the parametric pdf of Y, i.e., $p(Y, X)$, is differentiable with respect to X, and using the Cauchy-Schwartz inequality ($E^2\{ZW\} \leq E\{Z^2\}E\{W^2\}$), the variance of the unbiased estimator is lower-bounded as

$$\sigma^2_X \geq \frac{1}{I_F}, \tag{2.20}$$

where I_F is called the Fisher information and is defined as

$$I_F = E\left\{\left|\frac{\partial \log p(Y,X)}{\partial X}\right|^2\right\} = -E\left\{\frac{\partial^2 \log p(Y,X)}{\partial X^2}\right\}. \tag{2.21}$$

When X is itself a random variable, the pdf $p(Y, X)$ in (2.20) will represent the joint pdf of X and Y, and the expected value is obtained over both X and Y [59].

In Chapter 5 we update the CRLB for the problem of binary distributed quantization and use the resulting bound to design the quantizers, as well as to compare with the final MSE performance.

2.3 Assignment Problem

The assignment problem is one of the fundamental combinatorial optimization problems. It translates into finding the minimum-weight perfect matching in a weighted bipartite graph. The assignment problem is usually described as follows:
Assume that there are N tasks and N agents, and the cost of assigning task b $1 \leq b \leq N$ to agent n $1 \leq n \leq N$ is $\Gamma_{b,n}$. The goal is to perform all tasks by assigning exactly one agent to each task in a way to minimize the total cost. In a particular assignment, for task b the agent n_b is selected, where $1 \leq n_b \leq N$. Notice that for two different tasks b and $c \neq b$ the agents n_b and n_c cannot be the same, i.e., $n_b \neq n_c$. Therefore, an assignment $\pi = n_1, n_2, \cdots, n_N$ is a permutation of the numbers 1 through N. Consequently, we have $N!$ assignments. In a linear sum assignment problem, it is assumed that the total cost of assignment $\pi = \{n_1, n_2, \cdots, n_N\}$ is the sum of the individual costs. Therefore, given an $N \times N$ cost matrix $[\Gamma_{b,n}]$, the objective is to find the assignment with minimum total cost $\mathcal{C}(\pi)$, i.e.,

$$\pi^* = \arg\min_{\pi} \mathcal{C}(\pi) = \arg\min_{\pi} \sum_{b=1}^{N} \Gamma_{b,n_b}. \tag{2.22}$$

In Chapter 5, once we have designed the N binary quantizers, we need to assign the N quantizers to the N sensors. In an inhomogeneous WSN, where sensors have different noise qualities the assignment problem becomes important for getting the optimum performance results.

An assignment problem can be viewed as a bijective mapping between two finite sets U and V [60]. Let $G = (U, V; W)$ be a bipartite graph with vertex sets $U = \{u_1, u_2, \cdots, u_N\}$ and $V = \{v_1, v_2, \cdots, v_N\}$, and edge set W. Every edge $(u, v) \in W$ has one vertex in U and the other vertex in V and there is a cost $\Gamma_{u,v} \geq 0$ associated with it. A subset $\mathcal{M}$ of W is called a "matching" if every vertex of G coincides with at most one edge from $\mathcal{M}$. If every vertex of G coincides with an edge of the matching $\mathcal{M}$, the matching $\mathcal{M}$ is called a "perfect matching". Thus, in a perfect matching $|\mathcal{M}| = N$, see Fig. 2.3. The objective of the assignment problem (2.22) translates to finding a perfect matching in G with minimum sum of the weights. This form of describing the assignment problem is used in Subsection 2.3.1 to explain the Hungarian algorithm, which gives an iterative method for finding the minimum-weight perfect matching in a bipartite graph.

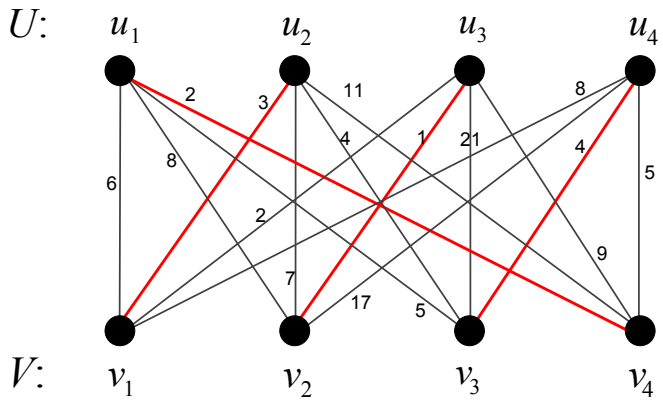

Figure 2.3: Minimum-weight perfect matching on a complete weighted bipartite graph.

2.3.1 Hungarian Algorithm

The Hungarian algorithm, first elaborated by Kuhn [61] and improved by Edmonds and Karp [62], provides an efficient algorithm for determining the minimum-weight perfect matching of a weighted bipartite graph. The Kuhn's method is based on finding the solution to a dual problem [60, 61]. The algorithm goes iteratively by

finding a partial solution based on the dual problem and improving the solution at every iteration. The dual problem can be described through the following definitions.

Definition 2.1. *A "feasible vertex labeling" is a function* $\ell : \{U, V\} \to \mathbb{R}$ *such that*

$$\ell(u) + \ell(v) \leq \Gamma_{u,v}, \qquad \forall u \in U, v \in V. \tag{2.23}$$

Definition 2.2. *The "equality graph" is the graph* $G_\ell = (U, V; W_\ell)$*, where*

$$W_\ell = \{(u, v) | \ell(u) + \ell(v) = \Gamma_{u,v}\} \tag{2.24}$$

The dual problem is constructed based on the following theorem

Theorem 2.1. *(Theorem 6 in [61])*
For any feasible labeling ℓ*, if* G_ℓ *contains a perfect matching* $\mathcal{M}$*, then* $\mathcal{M}$ *is a minimum-weight perfect matching of* G*.*

The proof is straightforward [61]. The algorithm starts by choosing an initial G_ℓ with few edges and finding the maximum matching on G_ℓ. At each iteration it grows G_ℓ to include more edges and consequently augments the matching, until the size of matching becomes equal to N. In order to describe how the algorithm augments a matching and grows G_ℓ we need to define some more terms.

Definition 2.3. *An edge in* W *is called a "matched" edge if it belongs to* $\mathcal{M}$*, and an "unmatched" edge otherwise. Likewise, a vertex is a matched vertex if it is an endpoint of a matched edge, and is an unmatched vertex otherwise.*

Definition 2.4. *A path in the graph whose edges are alternately matched and unmatched is called an "alternating path".*

Definition 2.5. *An "augmenting path" is an alternating path whose initial and terminal edges (and, hence, vertices) are unmatched.*

Definition 2.6. *An "alternating tree" rooted in a vertex* u *is a tree in which all paths emanating from* u *are alternating paths.*

The algorithm starts by constructing an initial equality graph $G_\ell^{(0)}$ based on an initial labeling $\ell^{(0)}$, and then choosing an initial matching $\mathcal{M}$ in $G_\ell^{(0)}$. Then it searches for an augmenting path for $\mathcal{M}$ in $G_\ell^{(0)}$. If such a path is found, an augmented (bigger size) matching is obtained by interchanging the matched and

unmatched edges along the alternating path. If no augmenting path exists the labeling and accordingly the equality graph are updated. The algorithm continues iterating this procedure until a perfect matching (size N) is found in the current equality graph. The method is shown in Algorithm 2. To find the augmenting paths in an equality graph an efficient algorithm called the Hopcroft-Karp algorithm can be used, which will be described in Subsection 2.3.2.

Algorithm 2 Hungarian Method for $G(U, V; W)$ and cost matrix $\{\Gamma_{u,v}\}$

Initialize the feasible labeling:

$$\begin{aligned} \ell^{(0)}(u) &= \min_v \Gamma_{u,v}, \quad \forall u \in U \\ \ell^{(0)}(v) &= 0, \quad \forall v \in V \end{aligned} \tag{2.25}$$

Determine the equivalent graph $G_\ell^{(0)}$ according to (2.24)
$i = 0$

while a perfect matching is not found **do**

Pick any matching $\mathcal{M}$ in $G_\ell^{(i)}$
while an augmenting path is found **do**
Find an augmenting path P with respect to $\mathcal{M}$
Augment $\mathcal{M}$ by swapping the role of unmatched and matched edges in P
(Other matched edges of $\mathcal{M}$ which do not lie on P remain matched edges)
end while

if $\mathcal{M}$ is not perfect **then**
$i = i + 1$
Find an unmatched vertex $u \in U$
Find an alternating tree rooted at u, partition its vertices to $\mathcal{S} \subset U$ and $\mathcal{T} \subset V$
Improve the labeling as:

$$\begin{gathered} \alpha = \max_{u \in \mathcal{S}, v \notin \mathcal{T}} (\ell(u) + \ell(v) - \Gamma_{u,v}) \\ \forall x \in \{U \cup V\} \quad \ell^{(i)}(x) = \begin{cases} \ell^{(i-1)}(x) - \alpha & \text{if } x \in \mathcal{S} \\ \ell^{(i-1)}(x) + \alpha & \text{if } x \in \mathcal{T} \\ \ell^{(i-1)}(x) & \text{otherwise} \end{cases} \end{gathered} \tag{2.26}$$

Determine the equivalent graph $G_\ell^{(i)}$ from $\ell^{(i)}$
end if

end while

2.3.2 Hopcroft-Karp Algorithm

The Hopcroft-Karp algorithm [63, 64] gives an efficient way to find a "maximum matching" (a matching that cannot be augmented) in a bipartite graph, through

finding augmenting paths. The complexity of the algorithm for a graph $G = (U, V; W)$ is $O(N^{2.5})$, where $N = |U| = |V|$. Instead of finding just a single augmenting path, the algorithm finds a maximal set of disjoint shortest augmenting paths with respect to a matching $\mathcal{M}$. The Hopcroft-Karp algorithm consists of two phases, a breadth first search (BFS) followed by a set of depth first searches (DFS) [65]. The BFS partitions the graph into layers by traversing the matched and unmatched edges alternatively at each layer. It start from the unmatched vertices in U and stops at finding the first set of unmatched vertices in V. For each unmatched vertex in V found through the BFS, the DFS computes an augmenting path from that vertex to an unmatched vertex in U. The Hopcroft-Karp algorithm is described in Algorithm 3. The disjoint augmenting paths that are found using the Hopcroft-Karp algorithm for the given $G_\ell^{(i)}$ are used in the Hungarian algorithm (Algorithm 2) to augment the matching $\mathcal{M}$. In Chapter 6, we use the Hungarian method and the Hopcroft-Karp algorithm to find the best sensor-quantizer assignment for the distributed quantization in WSNs.

Algorithm 3 Hopcroft-Karp Algorithm for $G_l(U, V; W_l)$ and a matching $\mathcal{M}$

Make the first layer $Ł_1 \subset U$ from the unmatched vertices in U
Traverse unmatched edges with an end point in $Ł_1$, put other endpoints in $Ł'_1 \subset V$.
$i = 1$

while no unmatched vertex in V is met **do**
  $i = i + 1$
  Traverse matched edges with an end point in $Ł'_{i-1}$,
  put the other endpoints (those not already met) into set $Ł_i \subset U$
  Traverse the unmatched edges with an end point in $Ł_i$,
  put the other endpoints (those not already met) into set $Ł'_i \subset V$.
end while

for each vertex $v \in Ł'_i$ **do**
  Go back through the layers in order $Ł_i, Ł'_{i-1}, Ł_{i-1}, \cdots, Ł_1$,
  pick one vertex from each layer at a time,
  by alternatively traversing the unmatched and matched edges
  (only pick vertices that are not already picked for other augmenting paths)
  Save the traversed path as the augmenting path for v.
end for

2.3.3 Conclusion

In this chapter, we provided the theoretical benchmark for the algorithms and solutions that are derived through the rest of this thesis. Specifically, the factor graph model presented in subsection 2.1.2 is used in Chapter 3 to model large multivariate,

nonlinear, estimation problems, and derive a distributed solution for them. This algorithm is a nonparametric solution standing on basis of Monte-Carlo integration, importance sampling, and Gibbs sampling methods presented in subsections 2.1.3 and 2.1.4. The fundamental theories of quantization presented in Section 2.2 are used in Chapters 4 and 5 of this thesis, where we solve the distributed quantization problem in a wireless sensor network. Finally, the Hungarian method and the Hopcroft-Karp algorithm described in Section 2.3 are used in Chapter 6 to develop an efficient solution for solving the bit-sensor assignment in inhomogeneous wireless sensor networks.

Chapter 3

Nonparametric Distributed Inference

In WSNs with no fusion centre many of the network tasks such as, decision making, hypothesis testing, detection, tracking, and etc, must be performed in a distributed fashion. These tasks are some sort of an inference or estimation problem. The existing optimal distributed estimation algorithms are usually not practical for wireless sensor networks, due to their high computational and data communication costs. Thus, suboptimal algorithms that use quantized data and are based on linear and Gaussian approximations have been proposed. Such approximations do not always work. Here, we propose a distributed estimation algorithm based on the well-known sum-product algorithm (SPA). To reduce computational complexity, we reformulate the sum-product update rules using particle filtering. We consider the problem of distributed target tracking based on quantized data in a WSN. By deriving the factor graph representation of this tracking problem, we apply our proposed algorithm and study its performance based on the number of quantization bits, the number of particles and the measurement noise.

3.1 Introduction

Recent improvements in software and hardware technology augments the employment of WSNs in different applications. In general, WSNs are used to monitor, detect or track an event or process. To fulfill these goals, sensors' local observation data is used to estimate or infer knowledge about some hidden (unobserved) variables. In recent years, various tools have been developed to help the general task of inference or estimation of hidden variables. Some of these tools have been used in

WSNs applications after being adapted to the nature of the specific estimation task. A powerful estimation tool which is commonly used to estimate the hidden variables based on the observed variables is the Kalman filter (KF) [42]. The optimum performance of the KF is obtained when the dependency between the hidden and observed variables are linear and the modeled noise is Gaussian. These conditions, however, do not always hold.

In a WSN, according to the limitations in the network resources, e.g. energy and bandwidth, it is desired to reduce the amount of intra-sensor communications. Therefore, the observation data is usually severely quantized to be represented with a few bits. Quantization violates the linear relations between the hidden and observed states, hence disqualifies the KF. [32] and [8] modify the conventional KF equations to handle quantized data instead of the precise measurement data. However, the exact analytical solution has high computational cost. Therefore, they approximate the non-Gaussian probability distributions with Gaussian distributions to derive feasible calculations. These approximate equations are close to the exact solution when the non-Gaussianity of a posteriori probability distributions is not severe. The method, hence, is limited to cases where Gaussian and linear assumptions are accurate.

In this chapter, we develop a new method for decentralized estimation of hidden variables from quantized sensors' observations in a WSN. For this purpose, we develop a stochastic inference algorithm based on factor graph (FG) modeling [47], a powerful tool to model the stochastic dependencies among the variables involved in an inference problem. Once the relations among the variables are derived as a FG, a message passing algorithm (MPA) is run to find an estimation about the unknown variables. The MPA determines the mechanism by which local messages are exchanged between the neighboring nodes on the graph. The FG and MPA become particularly useful in a WSN where we desire to break down the huge centralized computations into local computations at the sensors and small messages transmitted between neighboring nodes.

A famous MPA called the SPA is proposed by [47]. The FG and the SPA have been widely used together in many inference problems, including WSN applications. [66] uses FG modeling to derive algorithms for detection in multiple-input and multiple-output systems. [67] uses FG and the SPA to design a cooperative self tracking algorithm. [11] proposes a method based on SPA for link loss monitoring

in WSN, and [12] develops a link loss inference method for mobile ad hoc networks based on the SPA. [18] presents a method based on SPA for location estimation in mobile networks based on time-of-arrival measurements and [68] uses the FG and SPA to design a framework for cooperative localization using ultra-wideband ranging measurements in a wireless network.

The application of the SPA, however, is often restricted to problems involving binary or discrete variables or continuous variables with Gaussian distributions, because the SPA is often analytically intractable in problems where we have arbitrary functions as messages. In this chapter, we formulate a solution for the statistical estimation of continuous variables using FGs. We propose a non-parametric method for the general non-Gaussian and nonlinear inference problems. We use FG to model the stochastic relations between the hidden and observed variables. As for the MPA, we reformulate the sum-product update rules using PF and Monte Carlo method. In other words, using Monte Carlo method and importance sampling, we solve the analytical intractable integrals of the SPA for continuous variables. We combine two important tools, i.e. the SPA and the particle filtering, to enhance the functionality of both. By using particle-based MPA, we extend the ability of FG to efficiently cover many non-Gaussian, nonlinear frameworks with feasible complexity. Our proposed particle-based MPA is applicable to many estimation problems in WSNs, e.g. localization, tracking, link monitoring, etc. At each sensor, it performs local processing on the quantized data received from neighboring sensors to come to a general inference about the unknown variables.

The particle-based MPA is superior to the previous methods because it assumes no limitations on the type of relations between the variables, neither it restricts to Gaussian noise assumptions. Moreover, it is a low-cost algorithm in terms of communication costs, i.e., it reduces the energy consumption. The algorithm requires the sensors to broadcast only a low-bit quantized version of their data in their neighborhood, instead of the high-precision raw values. We have applied our algorithm for a problem of target tracking in a WSN. We achieved good results even when the number of quantized levels were small and the measurement noise was relatively large. Based on our experiments and comparison with some other applicable algorithms, our particle-based MPA is a reliable algorithm for low-cost distributed inference in WSNs.

In Section 3.2, we model the stochastic inference problems based on FG and the

SPA. In Section 3.3 we describe our particle filter based message passing algorithm (PF-based MPA) for continuous variables. In Section 3.4 target tracking problem in a WSN using quantized sensors data is solved by applying our particle-based message passing algorithm.

3.2 Factor Graph based Model

The FG framework [47] can be used to provide a graphical model for representing the inter-dependencies of the variables involved in a stochastic inference problem. The theoretical preliminaries are described in Section 2.1. The idea is to benefit from the conditional independencies inherited in the variables set to factorize a global pdf into local functions [1]. The FG representation of this factorization helps use the independencies in order to design distributed algorithms. A distributed/decentralized estimation algorithm is advantageous in WSNs where the observation data as well as the processing units are spread all over the network.

Assume that we are given a joint pdf of a number of hidden variables conditioned on a number of observed variables. Making use of the conditional independencies among the variables set, the conditional joint pdf can be factorized to smaller local functions and the factorization can be represented with a FG. In a FG, associated with each variable, there is a variable node x, and associated with each local function there is a function node f connected to its arguments, see Section 2.1.2. The FG framework is especially useful when we seek marginal distributions of some hidden variables conditioned on observed variables. For this, a MPA should operate on the FG. A famous MPA called the SPA introduced in [47] was described in Section 2.1.2. By running the SPA on a cycle-free FG, once a message is passed in both directions on every edge, the marginal a posteriori distribution of any variable can be found.

For a continuous random variable $x \in \mathbb{R}$, the message update rule of the SPA at a function node involves a multi-dimensional integral in place of a summation. Let us rewrite these rules for continuous random variables. Having $\aleph(x)$ indicate the set of neighbors of the variable node x and $\aleph(f)$ indicate the set of neighbors for the function node f, the update rules of the SPA are formulated as

[1] The FG framework is more general and can be used wherever a global function is factorized to local functions. In this thesis, however, we limit the discussions to pdfs and their factorizations to better connect to the estimation problem studied in the subsequent sections.

Update Rule 1:

$$\mu_{x\to f}(x) = \prod_{\lambda\in\aleph(x)\setminus f} \mu_{\lambda\to x}(x) \tag{3.1}$$

Update Rule 2:

$$\mu_{f\to x}(x) = \int_{\sim x} (f(X) \prod_{y\in\aleph(f)\setminus x} \mu_{y\to f}(y)) \tag{3.2}$$

In many problems, the local conditional distributions are not Gaussian and thus, we cannot work with simple mean and variance messages. One approach is to approximate the integrals with a Riemann sum. The draw back is the computational complexity which grows exponentially with the degree of the function nodes. In this chapter, we have developed a particle-based MPA for the general non-Gaussian, non-linear problems. The method is inspired by the concept of Monte Carlo integration whose complexity is independent of the order of the integral. Our algorithm is described in the next section.

3.3 Message Passing Based on Particle Filtering

To develop our PF-based MPA we make use of the Monte Carlo integration technique to solve the multi-dimensional integral in (3.2). Monte Carlo integration [43] is a probabilistic method used to approximate an integration, see Subsection 2.1.3 for theoretical details. Monte Carlo integration is the basis of PF [44] and [43]. PF has been widely used in a range of problems where the nonlinear and non-Gaussian properties of the variables disqualifies the application of solutions such as KF. The PF has been used in problems such as navigation, tracking and image processing [69]- [70]. [71] has also developed a non-parametric belief propagation method for a Bayesian network based on the concept of Monte Carlo integration, and [72] has used PF in belief propagation to solve the localization problem in sensor networks.

In this chapter, we formulate a message passing algorithm to be used with FGs. A major difference between this work and [71] and [72] is that our framework is developed to work on FGs with factor nodes of arbitrary degrees. However, the Bayesian network considered in [71] and [72] is equivalent to a FG with degree-two function nodes. In our proposed MPA, the messages, which are generally continuous pdfs, are represented by a set of $\mathcal{N}$ random samples and their importance weights

$\mu(x) = \{x^i, w^i; 1 \leq i \leq \mathcal{N}\}$. Our goal, therefore, is to formulate update rules at variable nodes and function nodes of a FG for such messages. In other words, the question is how to update x^i and w^i according to the SPA.

In order to formulate the update rule at a function node, we use the Monte Carlo integration concept to calculate the integrals. Assume that there are $\mathcal{V}+1$ variable nodes connected to a function node f, Fig. 3.1. Recalling (3.2), we rewrite the outgoing message from the function node f to the $(\mathcal{V}+1)$th variable node $x_{\mathcal{V}+1}$ as

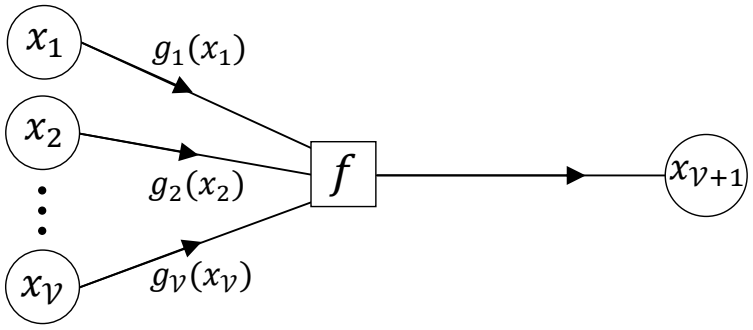

Figure 3.1: A function node in a factor graph.

$$\mu_{f\to x_{\mathcal{V}+1}}(x_{\mathcal{V}+1}) = h(x_{\mathcal{V}+1}) \quad = \int f(x_1, ..., x_{\mathcal{V}}, x_{\mathcal{V}+1}) g_1(x_1) ... g_{\mathcal{V}}(x_{\mathcal{V}}) dx_1 ... dx_{\mathcal{V}} \tag{3.3}$$

In the above equation $x_1, \ldots, x_{\mathcal{V}}$ are $\mathcal{V}$ variable nodes connected to the function node f, and $g_1(x_1), \ldots, g_{\mathcal{V}}(x_{\mathcal{V}})$ are their corresponding messages[2] toward f. Note that in statistical inference problems, the FG is derived from the stochastic factorization of an a posteriori joint pdf. Therefore, each message coming out of a variable node, e.g. $g_m(x_m); 1 \leq m \leq \mathcal{V}$, is a pdf conditioned on some observed parameters. While in a general FG, messages may not represent pdfs or conditional pdfs, viewing the messages as pdfs is instrumental to our PF-based MPA.

Defining vector $X = [x_1, ..., x_{\mathcal{V}}]^T$, the $\mathcal{V}$ dimensional integral of (3.3) can be written as a vector integral

$$h(x_{\mathcal{V}+1}) = \int f(x_{\mathcal{V}+1}, X) p(X) dX \tag{3.4}$$

where $p(X)$ can be viewed as the joint distribution of $x_1, \ldots, x_{\mathcal{V}}$, which according to (3.3) is equal to $\prod_{m=1}^{m=\mathcal{V}} g_m(x_m)$. Using the concept of Monte Carlo integration,

[2]Please note that, in general the function nodes and messages of a factor graph represent arbitrary functions and do not have to be pdfs. However, in stochastic inference problems and throughout this chapter functions f, g and h represent pdfs.

we can approximate the integral (3.4). The integral of (3.4) can be interpreted as the expected value of the function $f(X)$ with respect to X. The expected value can be numerically approximated through $\mathcal{N}$ i.i.d. samples of the random variable X drawn from its distribution $p(X)$. Therefore we can write

$$\begin{aligned} h(x_{\nu+1}) &= E_X\{f(x_{\nu+1}, X)\} \\ &\simeq \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} f(x_{\nu+1}, X^i), \quad \text{here} \quad X^i \sim p(X), \end{aligned} \tag{3.5}$$

where $X^i \sim p(X)$ means that the samples X^i are drawn from pdf $p(X)$. The notation $X^i \sim p(X)$ indicates that the sample X^i is drawn from the distribution function $p(X)$.

The above equation is a function of $x_{\nu+1}$ and can be regarded as the outgoing message from the function node f, i.e. $\mu_{f\to x_{\nu+1}}(x_{\nu+1})$. However, $h(x_{\nu+1})$ is a continuous function which should be represented as a particle-based message for the next stage of our PF-based algorithm. Therefore, we need to build the particle representation of $h(x_{\nu+1})$ as $\{x^i_{\nu+1}, w^i_{\nu+1}; 1 \le i \le \mathcal{N}\}$.

The best way to construct a particle representation is to directly draw $\mathcal{N}$ i.i.d. samples from $h(x_{\nu+1})$. However, it is not straightforward to sample from $h(x_{\nu+1})$ because it is a sum of $\mathcal{N}$ continuous functions. Hence, we use the importance sampling method [43] to construct the particle representation of the message $\mu_{f\to x_{\nu+1}}(x_{\nu+1})$. As described in [43], we need to select an importance distribution function $q(x_{\nu+1})$ which can be any positive function with the same support as $h(x_{\nu+1})$. The distribution $q(x_{\nu+1})$ is chosen such that it is straightforward to sample from. Then we draw $\mathcal{N}$ samples from $q(x_{\nu+1})$ and assign an importance weight to each sample, $w^i_{\nu+1}$. Thus,

$$\mu_{f\to x_{\nu+1}}(x_{\nu+1}) = h(x_{\nu+1}) \simeq \{x^i_{\nu+1}, w^i_{\nu+1}\} \quad 1 \le i \le \mathcal{N} \tag{3.6}$$

where

$$x^i_{\nu+1} \sim q(x_{\nu+1}) \qquad w^i_{\nu+1} \propto \frac{h(x^i_{\nu+1})}{q(x^i_{\nu+1})}$$

The importance weights should be normalized so that their sum is equal to one. Here, without loss of generality, we choose the importance distribution function to be $f(x_{\nu+1}, X^1)$. If it is not possible to sample from the function $f(x_{\nu+1}, X^1)$, one must choose another appropriate function as the importance density function and adjust the importance weights accordingly. Thus, to determine the ith particle and

its corresponding weight, we set $q(x_{\mathcal{V}+1}) = f(x_{\mathcal{V}+1}, X^1)$. By drawing the ith particle $x^i_{\mathcal{V}+1}$ from $f(x_{\mathcal{V}+1}, X^1)$, the ith importance weight will be proportional to the ratio of $h(x_{\mathcal{V}+1})$ to $f(x_{\mathcal{V}+1}, X^1)$ evaluated at $x^i_{\mathcal{V}+1}$. Therefore, the particle representation of $\mu_{f\to x_{\mathcal{V}+1}}(x_{\mathcal{V}+1})$ is

$$\mu_{f\to x_{\mathcal{V}+1}}(x_{\mathcal{V}+1}) \simeq \{x^i_{\mathcal{V}+1}, w^i_{\mathcal{V}+1}\} \qquad 1 \le i \le \mathcal{N} \tag{3.7}$$

where

$$x^i_{\mathcal{V}+1} \sim f(x_{\mathcal{V}+1}, X^1) \tag{3.8}$$

and

$$w^i_{\mathcal{V}+1} \propto \frac{\frac{1}{\mathcal{N}}\sum_{j=1}^{\mathcal{N}} f(x^i_{\mathcal{V}+1}, X^j)}{f(x^i_{\mathcal{V}+1}, X^1)} \tag{3.9}$$

The procedure of the particle-based message processing at a function node is represented in Algorithm 4. Note that any of the functions $f(x_{\mathcal{V}+1}, X^i)$ where $1 \le i \le \mathcal{N}$ can be chosen to be the importance function in (3.7). In a more general FG setup, when the messages are not pdfs, our approach is still applicable as long as the importance function is chosen to be a valid pdf and $\sum_{j=1}^{\mathcal{N}} f(x^i_{\mathcal{V}+1}, X^j)$ is always positive. For pdfs this condition is clearly satisfied.

As indicated in Algorithm 4, we need $\mathcal{N}$ (weighted) samples of X. According to (3.3) the product of $\mathcal{V}$ incoming messages to f construct the distribution of X. Thus, X can be viewed as an ordered $\mathcal{V}$-tuple with independent entries. A sample of X, therefore, can easily be constructed by putting together samples from $x_1, \ldots, x_m$. In our setup, however, samples are weighted. If all samples are equally weighted (i.e., all messages are represented by equally weighted particles), taking one particle from each message at a time, we can build a particle for X. When particles have unequal weights, we first use a resampling procedure to extract $\mathcal{N}$ equally weighted particles, then continue with the normal procedure. This resampling algorithm (Algorithm 5) draws i.i.d samples from a set of particles with nonequal weights. Notice that, if $\mathcal{V} = 1$ then X will be the same as x_1. In this case, there is no need for resampling. We can simply choose the sample set of X as $\{X^i, w^i\}_{i=1}^{\mathcal{N}} = \{x^i_1, w^i_1\}_{i=1}^{\mathcal{N}}$.

At a variable node we use the method developed in [51]. Assume that $\mathcal{F}+1$ function nodes, $f_1, \ldots, f_{\mathcal{F}+1}$, are connected to a variable node x, as shown in Fig. 2.2. The lth incoming messages is presented as $h_l(x) = \{x^i_l, w^i_l\}, 1 \le i \le \mathcal{N}$. To compute the particle representation of the outgoing message form the variable node $\mu_{x\to f_{\mathcal{F}+1}}(x)$, we use the Gibbs sampling method described in Section 2.1.4.

Algorithm 4 Message Passing at a Function Node

Having $\mathcal{V}$ messages from the nodes $x_1, x_2, \ldots, x_\mathcal{V}$,
Let the mth message, $1 \leq m \leq \mathcal{V}$, be represented by the particle set $\{x_m^i, w_m^i\}_{i=1}^{\mathcal{N}}$
for $m = 1$ to $\mathcal{V}$ **do**
  if all w_m^i are not the same for $1 \leq i \leq \mathcal{N}$ **then**
    Draw $\mathcal{N}$ new equally weighted particles using Algorithm 5
  end if
end for
for $i = 1$ to $\mathcal{N}$ **do**
  $X^i = [x_1^i, x_2^i, \ldots, x_\mathcal{V}^i]^T$
end for
for $i = 1$ to $\mathcal{N}$ **do**
  Sample a particle from the importance distribution

$$x_{\mathcal{V}+1}^i \sim f(x_{\mathcal{V}+1}, X^1)$$

  Assign the particles weight

$$w_{\mathcal{V}+1}^i = \frac{\frac{1}{\mathcal{N}} \sum_{j=1}^{\mathcal{N}} f(x_{\mathcal{V}+1}^i, X^j)}{f(x_{\mathcal{V}+1}^i, X^1)}$$

end for
Calculate the weights sum $W = \sum_{j=1}^{\mathcal{N}} w_{\mathcal{V}+1}^j$
for $i = 1$ to $\mathcal{N}$ **do**

$$w_{\mathcal{V}+1}^i = \frac{w_{\mathcal{V}+1}^i}{W}$$

end for

Algorithm 5 Resampling

Having the particle set $\{x^i, w^i\}_{i=1}^{\mathcal{N}}$
Construct the cumulative sum of weights as follows.

$$C(i) = \sum_{k=1}^{i} w^k \qquad 1 \leq i \leq \mathcal{N}$$

for $n = 1$ to $\mathcal{N}$ **do**
  Take a sample from the uniform distribution $[0, 1] \rightarrow u$.

  $j = C^{-1}(u)$

  Let the nth particle and weight be $\{x^n, w^n\} = \{x^j, w^j\}$

end for

3.4 Target Tracking in Wireless Sensor Networks

3.4.1 System Model

In this section we focus on the problem of target tracking in a wireless sensor network. The target is a moving object which is monitored at each time interval k, by $\mathcal{S}_k \geq 3$ number of sensors . We assume that the sensors know the location of themselves and all the other sensors in the surveillance area which is a common assumption, e.g., [32], [8] and [9]. Also, We assume that data transmission is not subject to errors.

The sensors quantize their measurements using a L-level quantizer and transmit the quantized data bits over the wireless channel to their neighboring nodes. Each node receives quantized measurement data from its neighbors. Then, it runs a local tracking algorithm using its own measurement as well as the received data from its neighbors to estimate the location of the target at time step k.

At the end of time step k, each sensor in the target area [3] has a local estimation of the target's state. Therefore, each sensor in the target area can identify the closest sensor to the target based on its own estimation of the location of the target and the sensors' locations information. The sensor who finds itself the closest sensor to the target broadcasts its estimation so that the sensors involved in the next step of the estimation process have some knowledge about the target's state at the previous step. We choose the closest sensor to broadcast its estimation because it is more likely to have the best estimation of the target's state.

Due to inaccuracy of distance measurements the sensors' estimations are not perfect. Therefore, by acting based on the above autonomous algorithm it is possible that no or more than one sensor broadcast their estimation. If more than one sensor transmits, one can use only one of the received estimated data to start the next step. On the other hand, to avoid no transmission, when sensors are listening to the channel for data, if nothing is heard for a while, the sensor(s) who thinks is the second closest neighbor is programmed to broadcast its data. Please notice that for this part of the algorithm, other simple solutions can be suggested. For more studies on how to choose the best sensor at each step see [73].

To develop the target tracking algorithm, first we have to model the problem. The process who is being tracked is the dynamics of a moving target. We consider a

[3] A sensor is in the target area if it can have a measurement from the target.

4-dimensional state vector, X, to describe the dynamics of the target and use a first order Markov chain to model it as a random process. Assume that the 2-D position of the target at time step k is defined as $P_k = [x_k \ y_k]^T$, and its velocity is defined as $V_k = [vx_k \ vy_k]^T$. The target will also have an acceleration, $A_k = [ax_k \ ay_k]^T$, whose elements take a discrete value from the set $\{0, -g, +g\}$. The location at time step k will be related to the location at step $k-1$ according to the following state equation.

$$X_k = \begin{bmatrix} x_k \\ y_k \\ vx_k \\ vy_k \end{bmatrix} = F \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ vx_{k-1} \\ vy_{k-1} \end{bmatrix} + G \begin{bmatrix} ax \\ ay \end{bmatrix} + G \begin{bmatrix} u_x \\ u_y \end{bmatrix} \tag{3.10}$$

where

$$F = \begin{bmatrix} 1 & 0 & t_s & 0 \\ 0 & 1 & 0 & t_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} G = \begin{bmatrix} t_s^2/2 & 0 \\ 0 & t_s^2/2 \\ t_s & 0 \\ 0 & t_s \end{bmatrix}$$

and t_s is the step size, $[u_x \ u_y]^T$ is the process noise. Each of the two components of the acceleration vector at time step k, is either $0, -g$ or $+g$ with probabilities modeled as a random Markov jump with the initial probability vector Pr_0 and the transition matrix Tr.

The next step is to define a measurement model which relates the measurements data to the target states. In our tracking scenario, the sensors can only measure their distance to the target. Having only distance measurements, we need at least three measurement values to infer a 2-D location. Our algorithm can incorporate the measurements of more than one sensor node to find the 2-D location from the distance observation data. Let $[n_x \ n_y]^T$ define the location of a sensor in a 2-D plain then, the measurement model will be

$$z_k = \sqrt{(x_k - n_x)^2 + (y_k - n_y)^2} + v_k, \tag{3.11}$$

where v_k is the measurement noise which stands for the error in measuring the distance. Notice that at time step k, there are $\mathcal{S}_k$ sensors measuring the distance of the target. Therefore, we have $\mathcal{S}_k$ measurement equations similar to (3.11) with different n_x and n_y values. For the ease of notations, we have ignored the sensor index in (3.11) and parts of the subsequent discussions.

3.4.2 Solution Setup

As mentioned earlier, in WSN applications, sensor nodes are energy constrained. Therefore, it is desired to transmit fewer data bits to reduce energy and bandwidth usage. For this purpose, we quantize the high precision sensors measurements prior to sending them on the channel. One way to do this, is to quantize the absolute z_k. This is not very efficient because z_k ranges from zero to very large numbers and hence, the quantization requires many bits to convey enough information. A better way is to quantize a relative value instead of the absolute value. In our method we quantize the normalized innovation data which is defined as

$$\tilde{\mathrm{I}}_k = \frac{z_k - \hat{z}_{k-1}}{N_f} \tag{3.12}$$

where

$$\hat{z}_{k-1} = \sqrt{(\hat{x}_{k-1} - n_x)^2 + (\hat{y}_{k-1} - n_y)^2}.$$

Also, N_f is a normalization factor which estimates the maximum value of $z_k - \hat{z}_{k-1}$ and can be found to be

$$N_f = t_s V_{max} + \frac{t_s^2}{2} g + 5\frac{t_s^2}{2}\sigma_{v_k}. \tag{3.13}$$

Here σ_{v_k} is the measurement noise standard deviation and V_{max} is a rough estimation of the maximum velocity of the target.

Having the measurement at time k, each sensor calculates the innovation value using (3.12). According to the definition of N_f, almost always $\tilde{\mathrm{I}}_k \in [-1, 1]$. Thus, to quantize $\tilde{\mathrm{I}}_k$, we use a L-level quantizer which quantizes the range $[-1, 1]$ to L levels. This quantized value is referred to as q_k. In rare cases that $\tilde{\mathrm{I}}_k$ is outside the range $[-1, 1]$, it is truncated to -1 or $+1$ accordingly.

Our goal is to infer the state of the target, X_k, at each time step k from the quantized innovation data of $\mathcal{S}_k \geq 3$ sensors. We restate the problem as finding the a posteriori pdf of X_k conditioned on the previous target state and acceleration. In other words, we are looking to find the marginal distribution

$$f(X_k, z_k, q_k | X_{k-1}, \hat{X}_{k-1}, A_{k-1}). \tag{3.14}$$

We assume that at time step k we already have the a posteriori pdf of X_{k-1} from which we decide the estimated state $\hat{X}_{k-1}$. The joint conditional pdf of (3.14) can be factorized It must be mentioned that the choice of the a posteriori pdf, as well

as the factorization process determines the FG and in turn the MPA calculations. If the factorizations ends up with more but smaller factors the representing FG will have more function nodes but with smaller degrees. Note that degree of a function node determines the integral dimension in the message update rule. Therefore, if a function cannot be further split into smaller factors the message update rule at the corresponding node involves a higher dimensional integral. This is where our PF-based MPA becomes specifically beneficial, since owing to the Monte Carlo method the complexity of our message update rule will be independent of the node degree.

Factorization of (3.14) can be done based on the stochastic relations of the variables inferred from (3.10), (3.11) and (3.12), and by using the probability rules. First, using the Bayes' rule, the joint a posteriori pdf can be written as

$$f(X_k, z_k, q_k|X_{k-1}, \hat{X}_{k-1}, A_{k-1}) = \\ f(z_k, q_k|X_k, X_{k-1}, \hat{X}_{k-1}, A_{k-1}) f(X_k|X_{k-1}, \hat{X}_{k-1}, A_{k-1}). \tag{3.15}$$

The second factor in the right hand of (3.15) can be simplified to $f(X_k|X_{k-1}, A_{k-1})$. This is justified because having X_{k-1}, knowledge of $\hat{X}_{k-1}$ will not give more information about X_k. The first factor in (3.15) can be again broken into two factors using the Bayes' rule. Hence, the joint a posteriori can be written as

$$f(X_k, z_k, q_k|X_{k-1}, \hat{X}_{k-1}, A_{k-1}) = \\ f(q_k|z_k, X_k, X_{k-1}, \hat{X}_{k-1}, A_{k-1}) f(z_k|X_k, X_{k-1}, \hat{X}_{k-1}, A_{k-1}) f(X_k|X_{k-1}, A_{k-1}). \tag{3.16}$$

According to (3.12) q_k is only a function of z_k and $\hat{X}_{k-1}$. Also, according to (3.11) z_k is only dependent on X_k. Therefore,

$$f(X_k, z_k, q_k|X_{k-1}, \hat{X}_{k-1}, A_{k-1}) = \\ f(X_k|X_{k-1}, A_{k-1}) f(z_k|X_k) f(q_k|z_k, \hat{X}_{k-1}). \tag{3.17}$$

According to the factorization in (3.17), the factor graph associated with the problem is sketched in Fig. 3.2. We now apply the PF-based MPA introduced in Section 3.3 to the factor graph of Fig. 3.2. The details of the message processing rules at each function node and variable node of the graph are explained in the next subsection.

3.4.3 Particle-Based Message Passing Algorithm

The factor graph of Fig. 3.2 describes the inference problem at step k. The nodes X_{k-1} and $\hat{X}_{k-1}$ are also connected to the FG of the step $k-1$. At step k, the

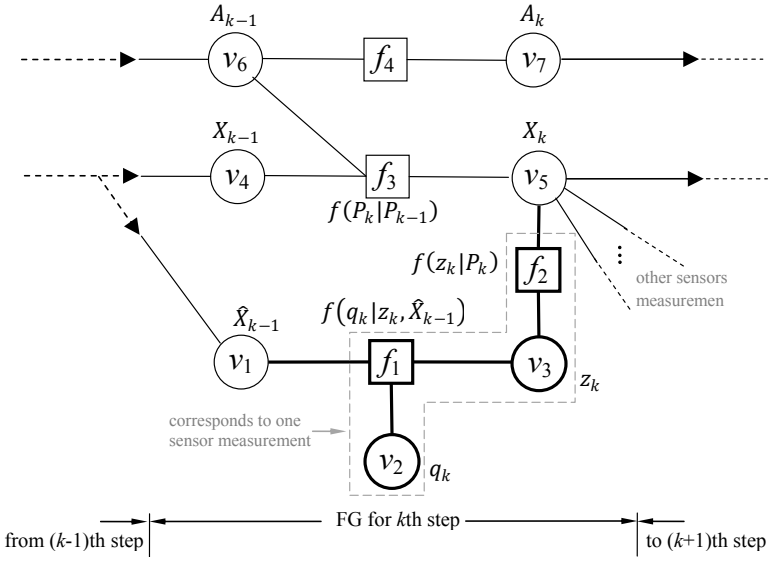

Figure 3.2: Factor graph of the tracking problem in WSN.

message passing starts from X_{k-1} and $\hat{X}_{k-1}$ whose messages are ready from the previous time step. More specifically, at step k, we know the value of $\hat{X}_{k-1}$ and the a posteriori pdf of X_{k-1} which is represented by $\mathcal{N}$ particles and their importance weights in the form of $X_{k-1} \simeq \{x^i, w^i; 1 \leq i \leq N\}$. To find the marginal distribution of X_k, we use the MPA described in Section 3.3.

The overall steps of the tracking method is given in Algorithm 6, where the MPA of Section 3.3 is used at Step 5. In the following, we describe all message update rules for this step.

Algorithm 6 Target Tracking in WSN based on Particle-based MPA

1. Mean and covariance of the final particle message at v_5 is broadcasted by the active sensor at step $k-1$ and is received by the sensors in the target area.
2. Each of the $\mathcal{S}_k$ sensors in the target area measures its distance from the target $z_k^j, 1 \leq j \leq \mathcal{S}_k$.
3. Each of the $\mathcal{S}_k$ sensors computes its innovation data, $\tilde{\mathrm{I}}_k^j, 1 \leq j \leq \mathcal{S}_k$.
4. Each of the $\mathcal{S}_k$ sensors quantizes their normalized innovation data to q_k^j and sends q_k^j to its neighbors.
5. Each of the $\mathcal{S}_k$ sensors runs a local version of particle-based MPA on a FG similar to the one shown in Fig. 3.2. The number of measurement branches of the FG at each sensor depends on the number of measurement data it receives from its neighbors.
6. At the end of step k, each of the $\mathcal{S}_k$ sensors have an estimation of X_k. The closest sensor to the target at this step broadcasts mean and covariance of its estimation.

The message going from variable node v_1 (Fig. 3.2) to the function node f_1 is

merely the value $\hat{X}_{k-1}$. Similarly, the message from variable node v_2 to the function node f_1, $\mu_{v_2 \to f_1}(q_k)$, is the value of q_k. Having the two incoming messages, f_1 can compute its outgoing message towards v_3. Since the two incoming messages are two fixed numbers, the outgoing message can easily be computed using the relations in (3.12) and the fact that q_k is the quantized version of $\tilde{\mathrm{I}}_k$.

Knowing q_k, the conditional pdf of $\tilde{\mathrm{I}}_k$ can be assumed to be uniform between $[q_k{-}\Delta/2 \quad q_k{+}\Delta/2]$, where Δ is the length of the quantization interval. The uniform assumption here is for simplicity. The actual conditional distribution depends on the marginal distribution of $\tilde{\mathrm{I}}_k$ and also the method of quantization. The distribution of z_k conditioned on q_k and $\hat{P}_{k-1}$ is easily found from (3.12) as below

$$f(z_k|q_k, \hat{z}_k) = \mathrm{U}(\hat{z}_k(1 + q_k - \Delta/2), \hat{z}_k(1 + q_k + \Delta/2)) \tag{3.18}$$

where $\mathrm{U}(a, b)$ represents a uniform distribution between a and b. The outgoing message from node v_3 to f_2 is the particle representation of the uniform distribution in (3.18). To obtain this representation, we sample $\mathcal{N}$ particles from (3.18) and assign equal weights. Thus

$$\mu_{v_3 \to f_2}(z_k) = \{z_k^i, w_z^i\} \quad 1 \leq i \leq \mathcal{N} \tag{3.19}$$

where for $\forall i$

$$z_k^i \sim \mathrm{U}\left(\hat{z}_k(1 + q_k - \Delta/2), \hat{z}_k(1 + q_k + \Delta/2)\right)$$

and $w_z^i = \frac{1}{\mathcal{N}}$. Note that the particles have equal importance weights because the samples have been drawn directly from the desired pdf.

At f_2, the incoming message to the function node is $\mu_{v_3 \to f_2}(z_k)$ represented in (3.19) and the local function at f_2 is $f(z_k|X_k)$ according to the factorization in (3.17). The stochastic dependence between z_k and X_k is inferred from the measurement equation (3.11). Since the measurement z_k is only affected by the target's location, the message from f_2 to v_5 is a function of only P_k and not the whole vector X_k. In other words, the local function $f(z_k|X_k)$ reduces to $f(z_k|P_k)$ and therefore, at f_2 only the particles of P_k will be generated and sent toward v_5.

Based on the particle-based update algorithm described in Algorithm 4, for each sample z_k^i from the message $\mu_{v_3 \to f_2}(z_k)$ we have to draw a sample of $P_k = [x_k \ y_k]^T$ from the local function $f(z_k^i|P_k)$. Noticing that all values of P_k are a priori equally likely, $f(z_k^i|P_k)$ is proportional with $f(P_k|z_k^i)$. Thus, it suffices to draw samples from $f(P_k|z_k^i)$. From (3.11) it is seen that the distribution of $P_k = [x_k \ y_k]^T$ conditioned

on z_k^i is on a circle with a random Gaussian radius $d = z_k^i + v_k$, centered at sensor location $[n_x \ n_y]^T$. Note that v_k is the measurement noise in (3.11). Therefore, $P_k = [x_k \ y_k]^T$ conditioned on z_k^i is described as

$$\begin{aligned} P_k &= \begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} d\cos(\theta) \\ d\sin(\theta) \end{bmatrix} + \begin{bmatrix} n_x \\ x_y \end{bmatrix} \\ \theta &\sim \mathrm{U}(0, 2\pi). \end{aligned} \tag{3.20}$$

According to (3.20) we can draw a sample for P_k for each sample of z_k. Finally the message $\mu_{f_2 \to v_5}(P_k)$ will be

$$\mu_{f_2 \to v_5}(P_k) = \{P_k^i, \tilde{w}^i\} \quad 1 \leq i \leq \mathcal{N} \tag{3.21}$$

where for $\forall i$

$$\begin{aligned} P_k^i &= \begin{bmatrix} x_k^i \\ y_k^i \end{bmatrix} = \begin{bmatrix} d^i\cos(\theta^i) \\ d^i\sin(\theta^i) \end{bmatrix} + \begin{bmatrix} n_x \\ x_y \end{bmatrix} \\ d^i &\sim f_{V_k}(v_k) + z_k^i \qquad \theta^i \sim \mathrm{U}(0, 2\pi). \end{aligned} \tag{3.22}$$

Here, $f_{V_k}(v_k)$ is the pdf of v_k. Now, we compute the particle weights according to (3.7). Notice that, in this example, $\mathcal{V} = 1$ therefore the vector X is simply equal to z_k. For each i, $1 \leq i \leq \mathcal{N}$, the weight $\tilde{w}^i$ is computed as

$$\tilde{w}^i = \frac{\sum_{j=1}^{\mathcal{N}} w_z^j f(z_k^j | P_k^i)}{f(z_k^i | P_k^i)} = \frac{\sum_{j=1}^{\mathcal{N}} w_z^j f_{V_k}(z_k^j - d^i)}{f_{V_k}(z_k^i - d^i)}$$

The last part of the equation has been written using the fact that evaluating the local function $f(z_k|P_k)$ at the point (z_k^j, P_k^i) is the same as evaluation $f_{V_k}(\cdot)$ at $z_k^j - d^i$.

Note that this procedure is performed at each sensor node and for all the quantized measurements received from their neighbors. At each sensor node, there are $\mathcal{S}_k$ parallel copies of this procedure and there will be $\mathcal{S}_k$ number of $\mu_{f_2 \to v_5}(P_k)$, each of them corresponds to the measurement of one sensor.

By this end, we are done with the message passing calculations on the lower branch of the FG of Fig. 3.2 which corresponds to the measurement data. The upper branch in Fig. 3.2 represents the progress of target acceleration through time. Based on our earlier discussion in 3.4.1, the target's acceleration is modeled as a Markov jump. Therefore, one can easily find the pmf of the i.i.d random variables ax_k and ay_k by propagating the pmf of ax_{k-1} and ay_{k-1} using the transition matrix, Tr. The calculation at the function node f_4 would be $Pr_k = Pr_{k-1} \cdot Tr$. Let Pr_k have a

general form of $[p_{1_k}, p_{2_k}, p_{3_k}]$ then the pmf of either components of acceleration will be

$$f_k(a) = p_{1_k}\delta(a) + p_{2_k}\delta(a-g) + p_{3_k}\delta(a+g) \tag{3.23}$$

where $\delta(a)$ indicates the discreet delta function. The message going from variable node v_6 to the function node f_3 is a set of $\mathcal{N}$ two dimensional vectors, $A^i_{k-1}, 1 \leq i \leq \mathcal{N}$, randomly taken according to the pmf $f_k(a)$. Note that, the two components of the acceleration, ax_{k-1} and ay_{k-1}, are assumed to be independent therefore, to sample a vector A^i_{k-1} we take two i.i.d samples from $f_k(a)$.

$$\mu_{v_6 \to f_3}(A_{k-1}) = \{A^i_{k-1}, w^i\}, \qquad 1 \leq i \leq \mathcal{N} \tag{3.24}$$

where

$$A^i_{k-1} = \begin{bmatrix} ax^i_{k-1} \\ ay^i_{k-1} \end{bmatrix}, \quad ax^i_{k-1} \text{ and } ay^i_{k-1} \sim f_k(a), \quad w^i = \frac{1}{\mathcal{N}}. \tag{3.25}$$

For the middle branch, which represents the evolution of the process, we have a root node v_4 representing X_{k-1}. The a posteriori pdf of X_{k-1} is approximated as $f(X_{k-1}) \simeq \{X^i_{k-1}, w^i\}, 1 \leq i \leq \mathcal{N}$ by taking $\mathcal{N}$ samples according to the estimated mean and covariance of X_{k-1} from the previous step. The same a posteriori will be transferred to the function node f_3, i.e. $\mu_{v_4 \to f_3}(X_{k-1}) = \{X^i_{k-1}, w^i\}$. At the function node f_3 we now have two incoming messages from which we have to find the outgoing message toward v_5. This message is calculated according to the particle-based message update given in Algorithm 4, with $\mathcal{V}$ equal to 2. After resampling the incoming message from v_4, we make the ith outgoing particle, X^i_k, by drawing a sample from $f(X_k|X^i_{k-1}, A^i_{k-1})$. According to (3.10) this sampling process will be

$$X^i_k = FX^i_{k-1} + GA^i_{k-1} + GU^i \tag{3.26}$$

where U^i is drawn from the process noise distribution. The weight for each particle X^i_k is computed according to Algorithm 4. Now we have the message toward v_5 as $\mu_{f_3 \to v_5}(X_k) = \{X^i_k, w^i\}, 1 \leq i \leq \mathcal{N}$.

At this point, we have $\mathcal{S}_k + 1$ messages at node v_5. One of them is from f_3, $\mu_{f_3 \to v_5}(X_k)$, which is a function of all the four components of the state vector. The other $\mathcal{S}_k$ messages are of the form $\mu_{f_2 \to v_5}(P_k)$ computed for each sensor's quantized measurement. These are only functions of the first two components of the state vector which are the position values. Now, at v_5, we multiply the first two components of all of the $\mathcal{S}_k+1$ messages using the Gibbs sampling algorithm of Section 2.1.4. For

the second and third components which correspond to the velocity V_k, we transfer the particles of the message $\mu_{f_3 \to v_5}(X_k)$ to construct the outgoing message from v_5.

3.4.4 Numerical Results

The PF-based algorithm is tested in a simulated target tracking scenario in a WSN. In our example, $\mathcal{S}_k = 5$ sensors are located inside an area of size 20×20 and every sensor can hear the other sensors' transmissions. At each time interval, all five sensors take a noisy measurement of the target's distance to themselves and broadcast the quantized version. The target is moving according to (3.10) where, $[u_x u_y]^T$ is a zero mean Gaussian noise with covariance matrix $\Sigma_d^2 = \begin{bmatrix} 0.02 & 0 \\ 0 & 0.02 \end{bmatrix}$ and also $t_s = 1$. The acceleration vector is a Markov jump process described in Section 3.4.1 with $g = 0.1$ and

$$Pr_0 = \begin{bmatrix} 0.7 & 0.15 & 0.15 \end{bmatrix} \quad \text{and} \quad Tr = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.4 & 0.5 & 0.1 \\ 0.4 & 0.1 & 0.5 \end{bmatrix}.$$

The measurement noise in (3.11) is also assumed to be zero mean Gaussian with variance σ_m^2. The algorithm, however, works with any other noise distribution. In the simulations, we have chosen σ_m proportional with the distance between the sensor and the target. Therefore, a sensor farther from the target has a noisier measurement of its distance than a closer sensor to the target. For a distance z, a signal-to-noise ratio (SNR) is defined as SNR$= \frac{z^2}{\sigma_m^2}$.

Since sensors have only distance measurements, at least $\mathcal{S}_k = 3$ sensors are required to find the target location. If at each time step more sensors can take measurements and communicate their quantized data to each other the algorithm will have better tracking performance. More number of sensors means having more instances of the graph shown in Fig 3.2, connected to each other at node v_5. Since we use Gibbs sampling at variable nodes, whose complexity is linear with respect to the number of Gaussian mixtures [51], the algorithm complexity will grow linearly with the number of sensors.

We compared the PF-based MPA with two other applicable algorithms. The first algorithm is the extended Kalman filter (EKF) which is fed by the quantized data as the measurement values. The second one is a Gaussian MPA which is run on the same FG as the FG in the PF-based MPA (Fig.3.2). In the Gaussian MPA, we approximate all conditional distributions of variable on the graph with Gaussian

distributions and therefore, replace all messages with their means and variances. The means and variances are analytically tracked. Gaussian approximations for MPA are common because of their low complexity. However, due to the nonlinear measurement equation (3.11), the nonlinear quantization noise and the target acceleration in the dynamic model in (3.10), the real distributions and therefore the actual messages are not Gaussian. It is also worth to mention that in the absence of these nonlinearities, the Gaussian MPA would reduce to an MPA representation of the KF on the associated FG [47] which would be the optimum solution.

We ran the three algorithms for a case with four level quantization and SNR= $50dB$. Fig. 3.3 depicts a sample outcome of target tracking with these algorithms. In this instance, the object is moving from top to the bottom of the $X-Y$ plane. Specifically, at around $(X, Y) = (13, 15)$, the other two methods lost track of the target while our algorithm followed the path correctly. This performance difference is related to the high nonlinearity of the target dynamic around that location. In order to have more reliable MSE values, we have run the three algorithms on 5 different target trajectories and averaged the location MSE over these runs. The average mean square of the location error (location MSE) for the EKF is 0.315, for the Gaussian MPA is 0.226 and for the PF-based MPA is 0.161.

Fig. 3.4 shows the performance of the PF-based MPA for $L = 4$ and different values of $\mathcal{N}$. It is seen that for $\mathcal{N} = 10$ or higher the algorithm achieves small error for location tracking. The result of an EKF is also depicted for comparison. The straight line shows the MSE of the EKF algorithm when there is no quantization in the measurement data.

Fig. 3.5 shows another comparison, where the non-quantized EKF is compared with our quantized PF-based MPA. It is seen that with 4-bit quantization or more the MSE of PF-based MPA is almost as good as the EKF without quantization. For very high quantization bits, the results of PF-based MPA is slightly better than EKF. This is because of the nonlinearity of (3.11) and also the presence of acceleration which is another source of nonlinearity.

The performance of PF-based MPA in low SNR is not as good as in high SNR. In fact, as SNR decreases the difference between the EKF curve and PF-based MPA curve in Fig. 3.4 and 3.5 increases. For example for SNR= 30 and $L = 4$ by increasing the number of particles, the MSE will improve from 0.776 for $\mathcal{N} = 3$ to 0.595 for $\mathcal{N} = 30$ and above. For this setup, MSE of the EKF without quantization

is 0.330 which is much better than PF-based MPA. It is also observed that for low SNR, increasing the number of quantization levels to more than 4 does not improve the performance. This is quite expected because when the actual measurements are very noisy, higher precision quantization (more bits) does not mean transmitting more information.

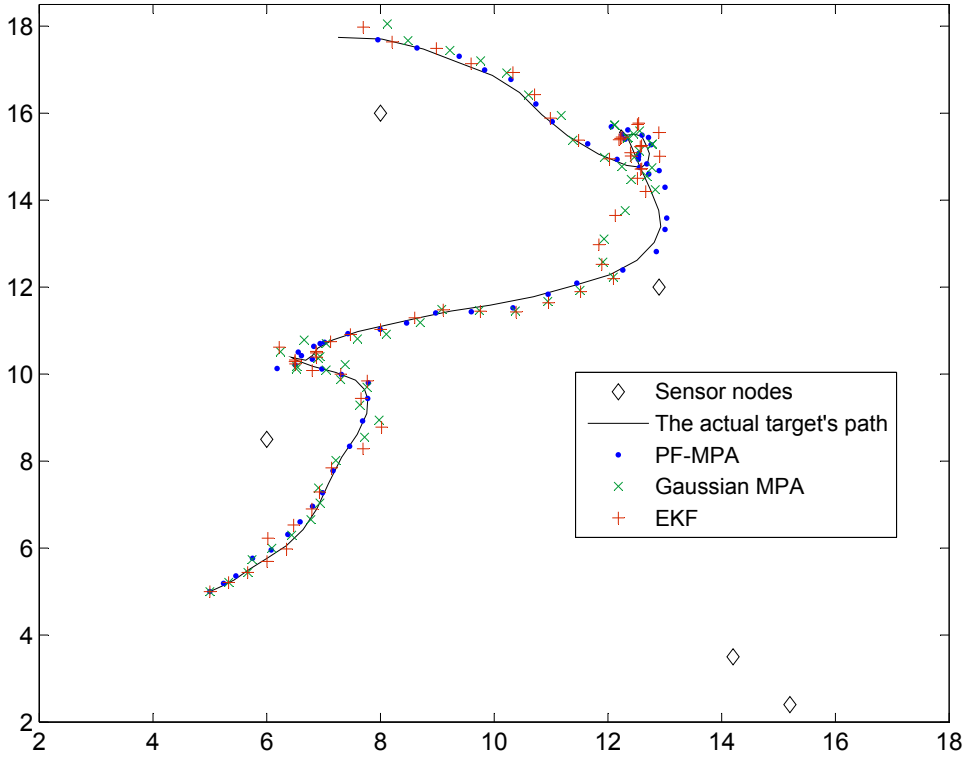

Figure 3.3: Result of tracking algorithm for $L = 4, \mathcal{N} = 40$, SNR$= 50dB$.

3.5 Data Association in Multi-Target Tracking

The second inference problem is the more complicated scenario of multi-target tracking which involves the data association problem [10, 74, 75]. In this problem there are a number of targets being tracked by a network of wireless sensors. At each time step, each sensor has some observations related to some of the targets. The challenging task is how to associate the measurements with the targets and track all targets simultaneously. Most of the existing solutions treat the data association task first and then apply conventional single-target tracking algorithms. Using the PF-based MPA, these two steps can be solved in a conjunctive fashion.

Since a 2-D tracking scenario has been discussed in Section 3.4, to simplify the

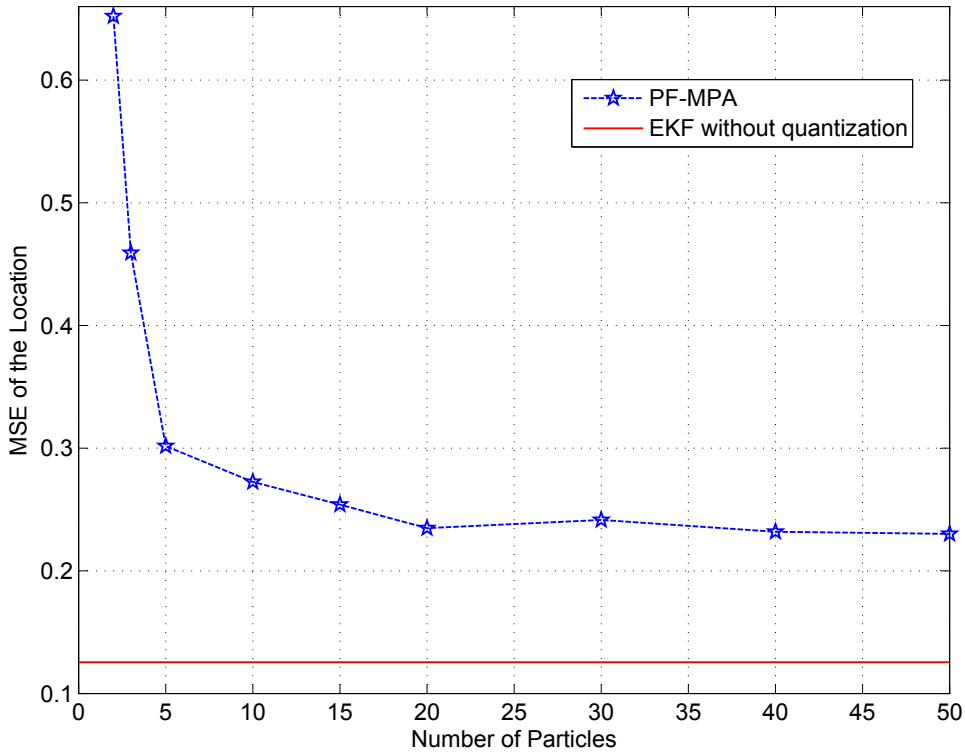

Figure 3.4: Performance of the algorithm versus the number of particles for $L = 4$, SNR$= 40dB$.

equations in this example we assume a 1-D trajectory for each target. We consider two targets being tracked by a single sensor. The extension of the algorithm to multiple sensors is straightforward and identical to the approach of Section 3.4. Assume the location of target 1 and 2, at time step k, is denoted by T_k^1 and T_k^2, respectively. At each time step, the sensor has two distance measurements from the targets, i.e., z^A and z^B, but it does not know which measurement is related to which target. Having the 2 measurement values, there are 2 association hypotheses; H0: z^A is related to the 1st target and z^B is therefore related to the 2nd target. H1: z^A is related to the 2nd target and z^B is therefore related to the 1st target. We define a hidden variable r which determines the association hypothesis. Therefore, r is a Bernoulli random variable that takes either the value 0, indicating H0, or 1, indicating H1. To finish the stochastic model setup, we also define z_k^1 and z_k^2 to be the expected measurement value for target 1 and 2, respectively. We can factorize

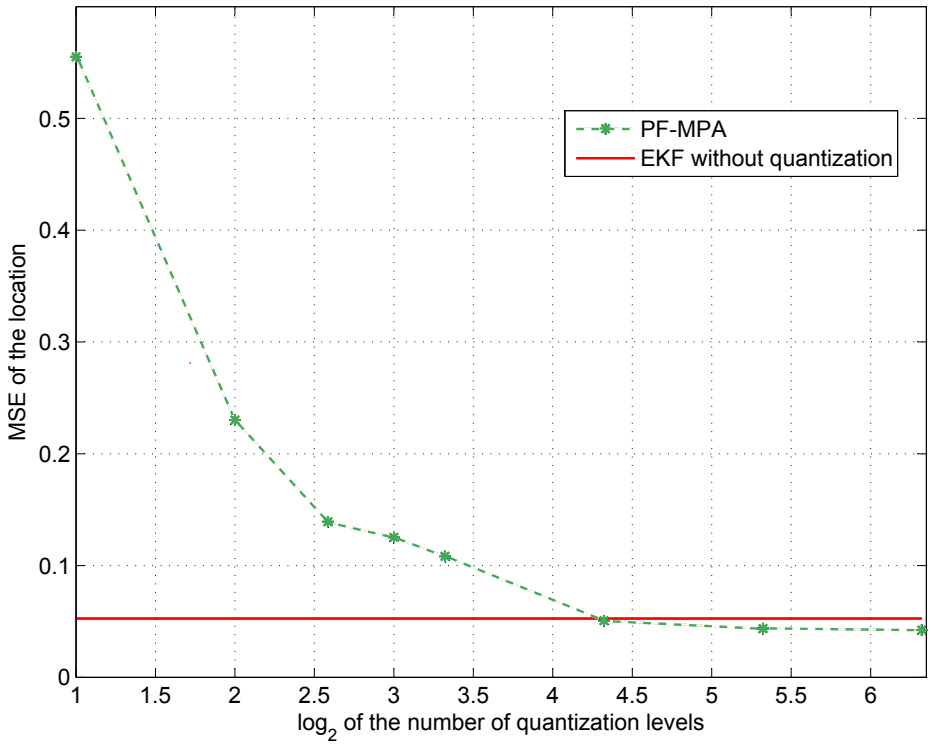

Figure 3.5: Performance of the algorithm versus log_2 of the number of quantization levels for $\mathcal{N} = 40$, SNR$= 50dB$.

the joint a posteriori pdf $p(T_k^1, T_k^2, z_k^1, z_k^2, r|T_{k-1}^1, T_{k-1}^2, z^A, z^B)$ as

$$p\left(T_k^1, T_k^2, z_k^1, z_k^2, r|T_{k-1}^1, T_{k-1}^2, z^A, z^B\right) =$$
$$p(r|z_k^1, z_k^2, z^A, z^B)p(z_k^1|T_k^1)p(z_k^2|T_k^2)p(T_k^1|T_{k-1}^1)p(T_k^2|T_{k-1}^2). \tag{3.27}$$

The associated factor graph is shown in Fig. 3.6. In the factorization (3.27), the factors $p(T_k^1|T_{k-1}^1)$ and $p(T_k^2|T_{k-1}^2)$ can easily be derived based on the dynamic model that we consider for target 1 and 2, respectively, similar to the approach in Section 3.4. Also, the factors $p(z_k^1|T_k^1)$ and $p(z_k^2|T_k^2)$ are decided based on the measurement models assumed for each target. The only remaining factor is $p(r|z_k^1, z_k^2, z^A, z^B)$ which basically is the probability of the hypothesis H0 or H1, knowing the values of the expected measurements z_k^1 and z_k^2, and the actual measurements z^A and z^B. An appropriate probability function must be chosen to represent $p(r|z_k^1, z_k^2, z^A, z^B)$, e.g.,

$$p(r|z_k^1, z_k^2, z^A, z^B) = \begin{cases} \tanh\left(K\mathcal{D}\right); & r = 0 \\ 1 - \tanh\left(K\mathcal{D}\right); & r = 1 \end{cases}, \tag{3.28}$$
$$\text{where} \quad \mathcal{D} = \frac{|z_k^1 - z^B||z_k^2 - z^A|}{|z_k^1 - z^A||z_k^2 - z^B|}$$

where K is a constant which is chosen to be 0.55 in our simulations. The PF-based MPA can now run on the factor graph of Fig. 3.6 to find the marginal a posteriori pdf of $T_1(k)$ and $T_2(k)$. Two sample results are shown in Fig. 3.7. In this example, both targets' model is a 1-D Markov chain with model noise variance of 0.0001. The measurement noise variance is 0.0001 for the top graph and 0.0004 for the bottom graph.

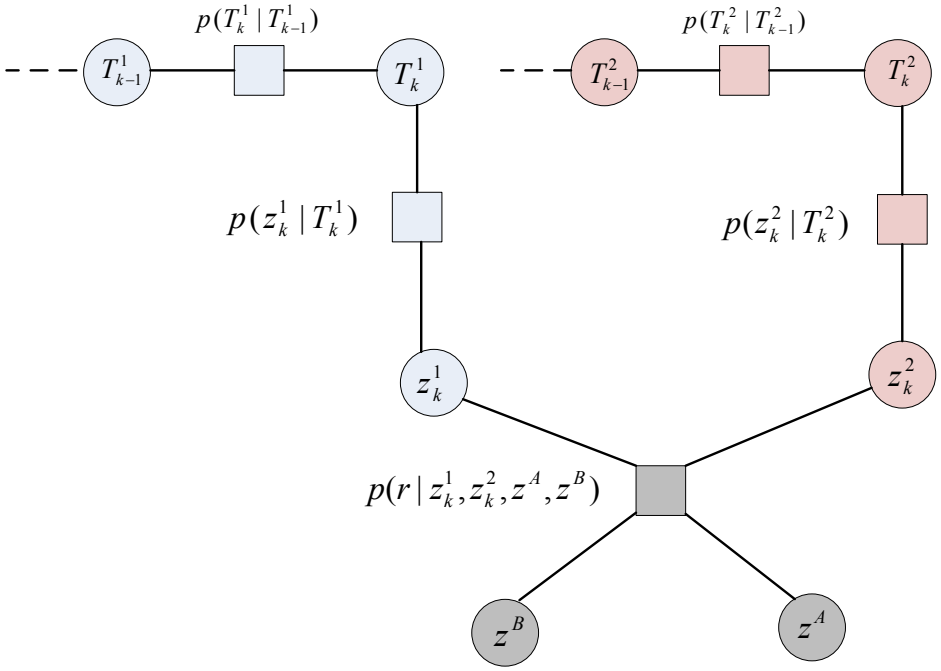

Figure 3.6: Factor graph of the multi-target tracking problem in a WSN.

3.6 Conclusion

In this chapter we introduced a particle-based message passing algorithm for general non-Gaussian, non-linear stochastic estimation problems. Specifically, for those inference problems that involve a large number of variables with arbitrary continuous pdfs analytical solutions are infeasible. Non-parametric solutions that are based on simple quantization and Riemann sum methods have high computational complexity which grow exponentially with the number of variables in the problem. Our particle-based message passing algorithm is based on Monte Carlo integration whose complexity is independent of the dimension of integrals. Hence, the complexity of our method does not exponentially increase by the number of variables.

An important application of such problems is the estimation task in WSNs. Our

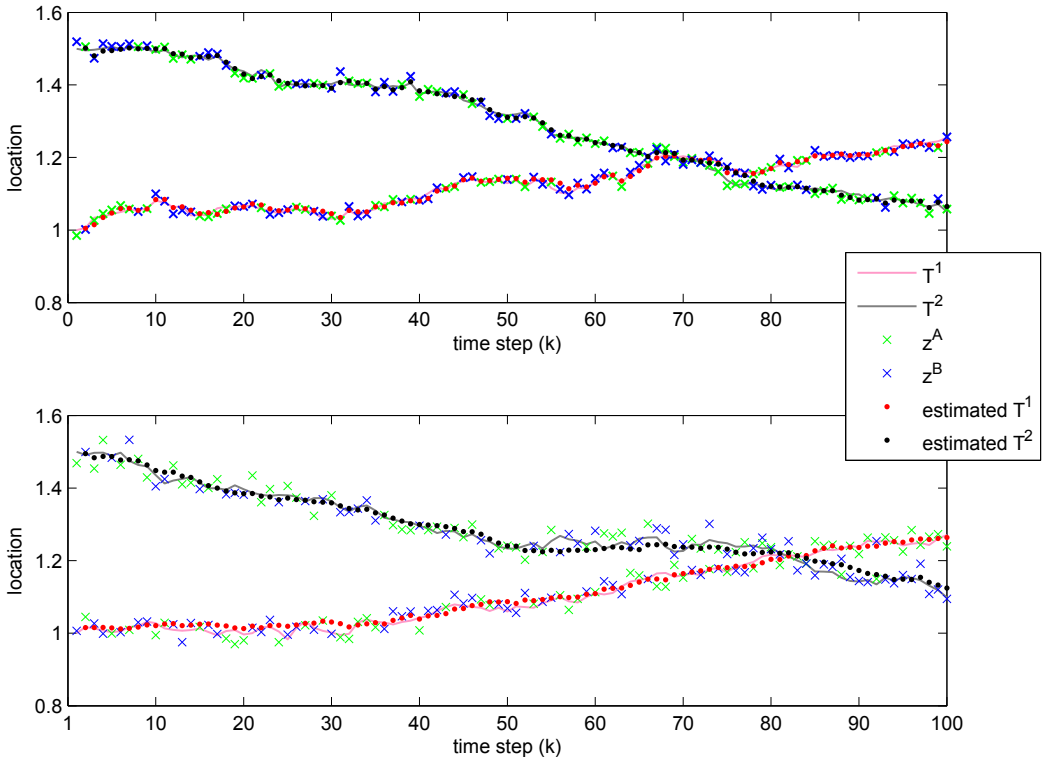

Figure 3.7: Sample results of multi-target tracking using PF-based MPA.

algorithm is suited to run in individual sensors in the WSN to do the distributed estimation process from the sensors' measurement data. It allows the sensors to work with low precision quantized data, which is less costly to send on the wireless channel. In the experiment of target tracking in a WSN, we achieved very good results by only using 3-bit quantized data. Unlike KF-based algorithms, our algorithm can work even when measurement data does not have a linear relation with the tracked process or when the noise is not Gaussian.

While we focused on a specific WSN problem, our methodology can be applied to other WSN estimation problems. Moreover, since we reformulated the sum-product update rules using particle filtering, our approach is readily applicable to problems with continuous variables, where the SPA is not analytically tractable. The studies in this chapter and the related algorithm and simulations are published in [49] and [50].

Chapter 4

Optimum Distributed Quantization

In distributed sensing systems with constrained communication capabilities, sensors' noisy measurements must be quantized locally before transmitting to the fusion center. When the same parameter is observed by a number of sensors, the local quantization rules must be jointly designed to optimize a global objective function. In this chapter we jointly design the local quantizers by maximizing the mutual information of the unknown parameter and the set of quantizers' outputs. An iterative approach is suggested for finding the local quantization rules. Using the mutual information as the design criterion, we can easily integrate the effect of communication channels in the design, and consequently design channel-aware quantization rules. We observe that the optimal design depends on both the measurement and channel noises. Moreover, our algorithm can successfully be used to design quantizers for different applications, such as estimation and detection. We demonstrate this through simulating an estimation and a detection application, where our method achieves comparable estimation and detection errors to the methods that have been specifically designed for those applications.

4.1 Introduction

A random source which has continuous amplitude requires infinite number of bits to be described. However, due to practical constraints in communication systems, e.g., limited storage or channel capacity, only finite number of bits can be accommodated. To compress a continuous-amplitude source X into limited amount of information (bits) its amplitude should be quantized to $Q(X)$, which takes values from a discrete

finite set. A side effect of this quantization is some loss of information about X, which depends on the quantizer's quality and the compression rate.

Quantization theory has been studied for long [23, 76]. The rate-distortion theory [53] describes the relation between the amount of distortion caused by the quantization, and the rate by which the quantized source can be presented. The theoretical limits described by the rate distortion function can only be asymptotically achieved by optimal source encoding.

The design of optimal quantizers cannot be achieved as a closed form solution for a specific source distribution. Lloyd and Max [22,23] propose an iterative algorithm for finding the best quantization rule for a random source to achieve the lowest distortion. The distortion measure they use is the MSE. Using the Lloyd-Max algorithm, the optimal[1] L-level quantization rule, which minimizes the estimation MSE can be found for a random scalar X with pdf $p(X)$. The joint quantization of multiple variables has been studied under vector quantization [56, 57].

A more interesting scenario is when the continuous-amplitude source is not observable and only a noisy version of it can be measured as $Y = X + W$, where W is some random noise. In this scenario, the observed quantity is quantized as $Q(Y)$; however, the goal is to achieve the best description of X. It has been shown in [24, 25, 77] that the optimal quantizer in this case can be achieved using the generalized Lloyd-Max algorithm, where the distortion measure is modified to include both the quantization and measurement noises.

A relatively recent and more challenging problem appears in distributed sensing systems, e.g., sensor networks. The problem can be described as distributed noisy source quantization. In a distributed sensing system, the same unknown source is observed by different measurement devices, each having a noisy observation as $Y_n = X + W_n$; $1 \leq n \leq N$, where W_n is the measurement noise of the nth observation. Each observation has to be quantized according to a local quantization rule $Q_n(Y_n)$ and sent to the FC. At the FC, the quantized values are used for estimation, detection, or classification purposes, based on the application of the distributed sensor network. In order to achieve an optimal solution, the local quantization rules have to be jointly optimized, making this problem more challenging than a centralized quantization. The rate-distortion bound is analytically untraceable in this case;

[1] It must be mentioned that all iterative algorithms for quantization design find a local optimal solution, which depends on the initial quantization rules.

however, upper and lower bounds have been derived in [55].

Designing the optimal quantizers for the above scenario has been considered by [26–29]. They suggest cyclic algorithms based on alternating minimization [30] to find the optimal N quantizers. The algorithm starts by initial guesses about the N quantizers, i.e., $Q_1^0, \cdots, Q_N^0$. During each iteration j, for each $1 \leq n \leq N$, the best quantization rule that optimizes a performance criterion (MSE [26, 27], Fisher information [28], or Ali-Silvey distances [29]) is found by fixing the other $N-1$ quantizers, and is assigned to Q_n^j. A heuristic method based on Fisher information is suggested in [33], which has lower complexity than optimal methods. For stringent capacity-constrained WSNs, binary quantization per sensor is suggested and some efficient techniques are proposed in [78–80].

Compared with the previous design algorithms for distributed quantization, i.e., [26–29], in this chapter we use the MI as the optimization measure for the distributed quantization design. We design quantizers that maximize the MI between the quantized data and the unknown parameter. Our motivation for using the MI is that it is a fundamental measure showing how much information one variable contains about another variable. We design for a set of quantizers that quantize the noisy measurements into a set of variables that together contain the most information about the unknown parameter.

The MI measure has the following benefits. It allows to design the quantizers independent of the choice of a decoder or estimator in the FC. Also, as we will discuss later, when using the MI measure the global optimization criterion can be broken down into smaller criteria. Finally, it allows to incorporate intermediate effects, such as the effect of the communication channels, in the design of optimal quantizers. Using the MI measure, we find optimal channel-aware local quantizers by integrating the channel in our design. To model the channel effect, the sensors' quantized data are mapped to binary codewords and each bit is sent over a binary symmetric channel (BSC). By maximizing the MI between the received data at the FC and the unknown parameter, we observe that depending on the channel noise the optimal quantizers can be different from the channel-unaware quantizers.

Designing distributed quantizers by maximizing the MI shows great results in various applications. This is evaluated for two applications, i.e., estimation and detection. We will show that the quantization rules obtained by maximizing the MI achieve the same performance as the quantizers specifically designed for the

estimation or the detection purpose.

This chapter is organized as follows. We first discuss the advantages of using the MI for the design algorithm, in Section 4.2. In Section 4.3, the problem is defined and formulated based on MI. Consequently, a design algorithm is devised in Section 4.4, assuming ideal communication channels. In Section 4.5, the algorithm is modified to include channel effect. Finally, the numerical results are presented and discussed in Section 4.6.

4.2 Mutual Information as the Optimization Criterion

Most optimal quantizer designs have used distortion measures, such as MSE [22–26, 77]. However, other measures have also been used as criteria to design quantizers, among them are the Ali-Silvey distances [29, 81, 82], Cramer-Rao lower bound and Fisher information [28,33,83,84]. One motivation for using these measures is the fact that they better fit some applications. For example, Ali-Silvey distance measures are shown to design quantizers that result in lower detection errors, which is useful for detection purposes [29, 81].

A fundamental measure, showing how much information about the unknown is conveyed in the quantized data, is the MI between the unknown and the quantized data. In this chapter, we base the design of the distributed quantizers on maximizing the MI. Specifically, we jointly design the quantizers so that the quantized data contain the highest information about the unknown. The MI criterion, to the best of our knowledge, has not been studied for distributed quantizer design in the literature.

A benefit of using the MI is to make the quantizer design independent of the estimation method or decoder. In design solutions based on distortion measures, such as squared-error or Hamming error [53], the estimation method is fixed, e.g., MMSE or ML, and the optimization of the quantizers is achieved depending on the estimator type. Using MI, the quantizer design stage becomes independent of the application. Depending on the application, an appropriate estimator/detector must be used in the FC, but this will not affect the quantizer design. This enables the designer to obtain a quantization algorithm useful for a range of applications such as estimation, detection, classification or feature extraction. Specifically for estimation purposes, the optimal quantizers designed based on minimizing the MSE (the Lloyd-Max algorithm) are those also with high MI [23,85]. This makes sense, because when

the quantized data carry more information about the unknown parameter, the FC has a better representation of the unknown; hence, it can estimate it more accurately. The successful performance of our MI-based algorithm in estimation and detection applications are shown in Subsections 4.6.1 and 4.6.2, respectively.

Using the MI measure in the distributed quantization design enables breaking down an N-sensor quantization problem into smaller problems. In fact, since the formula of the MI can be recursively broken down using the chain rule of MI, a simpler suboptimal solution can be derived by maximizing each component. Furthermore, the MI allows the design complexity to be significantly reduced when the sensors' measurements are conditionally independent. Such simplifications cannot be done when other measures, such as MSE, are used. This will be discussed in Subsection 4.4.2.

4.3 Problem Formulation Based on MI

The distributed quantization problem addressed in this chapter is defined as follows: Suppose X is a random scalar, which takes values in $\mathbb{R}$ with pdf $p(X)$[2]. A number of noisy measurements of X are observed at some distributed locations as $Y_n = h(X, W_n)$; $1 \leq n \leq N$, where W_n is the measurement noise, and h is an arbitrary function of variables X and W_n. The measurement noise at different sensors may be correlated. It is assumed that the distribution $p(Y_1, \cdots, Y_N|X)$ is known.

Due to communication constraints, the continuous-amplitude measurements have to be quantized before transmission. Therefore, Y_n is mapped to $Z_n \in \mathcal{L}_n = \{1, 2, \cdots, L_n\}$ using a local quantization rule $\mathcal{Q}_n : \mathbb{R} \to \mathcal{L}_n$, i.e., $Z_n = \mathcal{Q}_n(Y_n)$. A quantization rule $\mathcal{Q}_n$ is defined by a set of real-valued numbers called breakpoints, i.e., Γ_n, that divide $\mathbb{R}$ into partitions, and assign a value from $\mathcal{L}_n$ to each partition. Each quantized value Z_n, $1 \leq n \leq N$, is then transmitted over a communication channel, and the received symbol at the FC is called T_n . The complete problem model is shown in Fig 4.1.

Let $\{Y_1, \cdots, Y_N\}$, $\{Z_1, \cdots, Z_N\}$, and $\{T_1, \cdots, T_N\}$ be denoted by $Y_{1:N}$, $Z_{1:N}$, and $T_{1:N}$, respectively. Then the random variables X, $Y_{1:N}$, $Z_{1:N}$, and $T_{1:N}$ make a

[2]For the brevity of notations, we use the same symbol to address a random variable and its value.

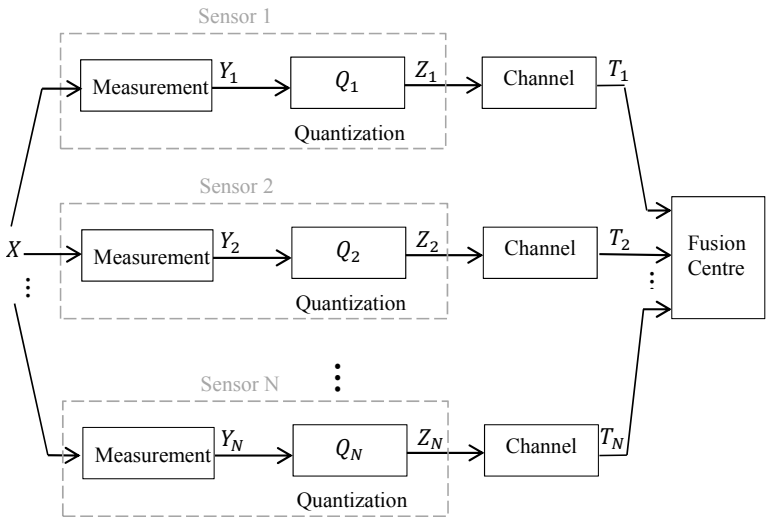

Figure 4.1: The complete model of the problem.

first-order Markov chain as $X \to Y_{1:N} \to Z_{1:N} \to T_{1:N}$, hence

$$\begin{aligned} &p(X, Y_{1:N}, Z_{1:N}, T_{1:N}) = \\ &p(X)p(Y_{1:N}|X)p(Z_{1:N}|Y_{1:N})p(T_{1:N}|Z_{1:N}). \end{aligned} \tag{4.1}$$

We will use this property to develop the design algorithm in Sections 4.4 and 4.5. We first consider ideal communication channels between the sensors and the FC, and derive an algorithm for designing the quantization rules in Section 4.4. In Section 4.5, we extend the algorithm to consider the channel effect and design channel-aware quantization rules.

From this point until Section 4.5, the communication channels are assumed ideal. Therefore, the goal is to derive $\mathcal{Q}_1(\cdot), \cdots, \mathcal{Q}_N(\cdot)$ so that on average, the random variables $Z_1, \cdots, Z_N$ are jointly a better representation of X. To achieve this goal, we maximize the MI between X and the N quantized data, as

$$\max_{\mathcal{Q}_1, \cdots, \mathcal{Q}_N} I(X; Z_1, Z_2, \cdots, Z_N). \tag{4.2}$$

Due to the step-wise characteristic of the quantization rules, finding an analytical solution to the problem in (4.2) is difficult. Therefore, in this chapter we find numerical methods to tackle this problem.

While we can develop numerical methods that directly work with (4.2), we find the following relaxation of (4.2) beneficial. As discussed in Subsection 4.4.2, this relaxation reduces design complexity, but as seen in Section 4.6, it does not cause any noticeable performance degradation. The main idea in this relaxation is to

benefit from the chain rule of MI [53]. The MI in (4.2) can thus be written as

$$I(X; Z_1, Z_2, \cdots, Z_N) = \sum_{m=1}^{N} I(X; Z_m | Z_{m-1}, \cdots, Z_1). \tag{4.3}$$

Hence, a suboptimal solution can be derived based on this recursive breakdown as

$$\begin{aligned}
&\mathcal{Q}_1^* = \arg\max_{\mathcal{Q}_1} I(X; Z_1) \\
&\mathcal{Q}_2^* = \arg\max_{\mathcal{Q}_2} I(X; Z_2 | Z_1) \\
&\vdots \\
&\mathcal{Q}_n^* = \arg\max_{\mathcal{Q}_n} I(X; Z_n | Z_1, Z_2, \cdots, Z_{n-1}) \\
&\vdots \\
&\mathcal{Q}_N^* = \arg\max_{\mathcal{Q}_N} I(X; Z_N | Z_1, Z_2, \cdots, Z_{N-1}),
\end{aligned} \tag{4.4}$$

where in the nth line, $1 \leq n \leq N$, $Z_1, \cdots, Z_{n-1}$ are quantized based on the previously found $\mathcal{Q}_1^*, \cdots, \mathcal{Q}_{n-1}^*$, respectively.

Finding the nth quantization rule, for $1 \leq n \leq N$, from (4.4) is less complex than finding all quantization rules from (4.2), see Subsection 4.4.2. In the following section, we develop a method to find a solution for the maximization problem in (4.4).

4.4 Optimum Quantizers for Ideal Channels

4.4.1 Algorithm

To find $\mathcal{Q}_n$, $1 \leq n \leq N$, one should solve the maximization in the nth line of (4.4). However, since $\mathcal{Q}_n$ is a discrete-valued function the optimization is not analytically traceable. In this section we provide a numerical method to find a local optimal solution for the nth quantization rule based on (4.4). Assume that $\mathcal{Q}_1^*, \cdots, \mathcal{Q}_{n-1}^*$ are known. To find the nth quantization rule $\mathcal{Q}_n^*$, according to (4.4) we need to maximize $I(X; Z_n | Z_1, Z_2, \cdots, Z_{n-1})$. When the $n-1$ previous quantization rules are fixed, this is equivalent to the following optimization (Note that $Z_{1:n} = \{Z_1, \cdots, Z_n\}$.)

$$\mathcal{Q}_n^* = \arg\max_{\mathcal{Q}_n} I(X; Z_{1:n}). \tag{4.5}$$

Using the chain rule of MI and the definition of MI based on the entropy [53],

$$\begin{aligned}
I(X; Z_{1:n}) =& I(X, Y_{1:n}; Z_{1:n}) - I(Y_{1:n}; Z_{1:n} | X) \\
=& H(X, Y_{1:n}) - H(X, Y_{1:n} | Z_{1:n}) - H(Y_{1:n} | X) + H(Y_{1:n} | X, Z_{1:n}).
\end{aligned} \tag{4.6}$$

In (4.6), the entropy terms involving only X and $Y_{1:n}$ are independent of the choice of quantizers. Therefore, we can reduce the optimization problem to[3]

$$\begin{aligned}\mathcal{Q}_n^* &= \arg\max_{\mathcal{Q}_n} \big[- H(X, Y_{1:n}|Z_{1:n}) + H(Y_{1:n}|X, Z_{1:n})\big] \\ &= \arg\max_{\mathcal{Q}_n} \int_X \int_{Y_{1:n}} \sum_{Z_{1:n}} p(X, Y_{1:n}, Z_{1:n}) \log p(X|Z_{1:n}) \\ &= \arg\max_{\mathcal{Q}_n} \int_X \int_{Y_{1:n}} \sum_{Z_{1:n}} p(X, Y_{1:n}) p(Z_{1:n}|Y_{1:n}) \log p(X|Z_{1:n}),\end{aligned} \tag{4.7}$$

where the last equation is the consequence of the Markov chain property in (4.1). We refer to this new objective function as $\mathcal{I}(X; Z_{1:n})$, i.e.,

$$\mathcal{I}(X; Z_{1:n}) = \int_X \int_{Y_{1:n}} \sum_{Z_{1:n}} p(X, Y_{1:n}) p(Z_{1:n}|Y_{1:n}) \log p(X|Z_{1:n}). \tag{4.8}$$

Note that maximizing $\mathcal{I}$ is equivalent to maximizing the MI in (4.5).

Since for each $1 \leq j \leq n$, the jth quantized value Z_j solely depends on the jth measurement Y_j, we have

$$p(Z_{1:n}|Y_{1:n}) = p(Z_1|Y_1) \cdot p(Z_2|Y_2) \cdots p(Z_n|Y_n).$$

Therefore, (4.7) can also be written as

$$\begin{aligned}\mathcal{Q}_n^* &= \arg\max_{p(Z_n|Y_n)} \mathcal{I}(X; Z_{1:n}) \\ &= \arg\max_{p(Z_n|Y_n)} \int_X \int_{Y_{1:n}} \sum_{Z_{1:n}} p(X, Y_{1:n}) \times p(Z_{1:n-1}|Y_{1:n-1}) p(Z_n|Y_n) \log p(X|Z_{1:n})\end{aligned} \tag{4.9}$$

Note that, in the above formula, the maximization is on $p(Z_n|Y_n)$. It is straightforward to see that the probability function $p(Z_n|Y_n)$ is just another form of defining the nth quantization rule. Since $\forall y \in \mathbb{R}$ the quantization rule maps y to a value $l \in \mathcal{L}_n$ such that $l = Q_n(y)$, we have

$$p(Z_n|Y_n = y) = \delta_{lZ_n}, \tag{4.10}$$

where function δ_{ab} is the Kronecker delta function, which is equal to one where $a = b$ and zero elsewhere. Note that in (4.9), $p(X|Z_{1:n})$ also depends on the quantization rule or $p(Z_n|Y_n)$.

To solve the optimization problem in (4.9), motivated by [86] we use the double maxima approach by converting (4.9) to a larger maximization problem. The maximization in (4.9) can be achieved through the next three steps, which is proven in Appendix A.

[3]We have dropped $dX, dY_1, \cdots, dY_n$ from all the formulas to save space.

i) The maximum of the objective function in (4.9), namely $\mathcal{I}^*$, can be written as

$$\mathcal{I}^* = \max_p \max_f \mathcal{I}(p, f) \tag{4.11}$$

where p is a short term for $p(Z_n|Y_n)$ and f is a short term for $f(X, Z_{1:n})$, a function of X and $Z_{1:n}$, such that for all realizations of $Z_{1:n} \in \mathcal{L}_1 \times \mathcal{L}_2 \cdots \times \mathcal{L}_n$, $\int_X f(X, Z_{1:n}) = 1$. And,

$$\mathcal{I}(p, f) = \int_X \int_{Y_{1:n}} \sum_{Z_{1:n}} p(X, Y_{1:n}) \times p(Z_{1:n-1}|Y_{1:n-1}) p(Z_n|Y_n) \log f(X, Z_{1:n}). \tag{4.12}$$

ii) Now, for a fixed p, $\mathcal{I}(p, f)$ is maximized by

$$f^*(X, Z_{1:n}) = \frac{\int_{Y_{1:n}} p(X, Y_{1:n}) p(Z_{1:n}|Y_{1:n})}{\int_X \int_{Y_{1:n}} p(X, Y_{1:n}) p(Z_{1:n}|Y_{1:n})}. \tag{4.13}$$

iii) And, for a fixed f, p^* that maximizes $\mathcal{I}(p, f)$ is obtained as

$$\begin{aligned} &\forall y \in \mathbb{R} \qquad p(Z_n|Y_n = y) = \delta_{l^* y}, \\ &\text{where} \qquad l^* = \arg\max_l \int_X \int_{Y_{1:n-1}} \sum_{Z_{1:n-1}} p(X, Y_{1:n-1}, y) \\ &\qquad\qquad \times p(Z_{1:n-1}|Y_{1:n-1}) \log f(X, Z_{1:n-1}, l), \end{aligned} \tag{4.14}$$

where $l \in \mathcal{L}_n = \{1, 2, \cdots, L_n\}$.

It is shown in Appendix A that these procedures find $\mathcal{I}^*$. Using *i*, *ii* and *iii*, an iterative algorithm can be derived as Algorithm 7 to find the optimal nth quantization rule. In Algorithm 7, ϵ determines the condition to stop the iterations. Since the

Algorithm 7 Iterative algorithm to find nth quantization rule

$j = 0$
initialize the quantization rule: p^0
repeat
  $j = j + 1$
  maximize $\mathcal{I}(p, f)$ for fixed p, according to (4.13) or (4.20)

$$f^j = \max_f \mathcal{I}(p^j, f)$$

  maximize $\mathcal{I}(p, f)$ for fixed f, according to (4.14) or (4.21)

$$p^j = \max_p \mathcal{I}(p, f^j)$$

until $|\mathcal{I}(p^j, f^j) - \mathcal{I}(p^{j-1}, f^{j-1})| > \epsilon \mathcal{I}(p^j, f^j)$

objective function $\mathcal{I}(X; Z_{1:n})$ is increased at each iteration and is upper-bounded[4]

[4]This is because in (4.6), the MI (left-hand side) is upper-bounded and the first and third term on the right-hand side are fixed.

the algorithm converges to a local maximum, which is also a local maximum for $I(X; Z_{1:n})$. This can be also investigated in Fig. 4.2.

4.4.2 Complexity

All optimum quantizer design algorithms, e.g., [22, 23, 26, 28, 29, 87], have an iterative structure. At each iteration the objective function, which is a functional of the quantizer $Q(\cdot)$, is changed by a small amount until the change is less than a predefined threshold. At that point, the objective function is reached a local optimum and the optimum quantizer is found. In particular, for distributed quantizers design, where the objective function is a functional of $Q_1(\cdot), Q_2(\cdot), \cdots, Q_N(\cdot)$, each iteration involves calculating an N-dimensional integral [26, 28, 29].

Our algorithm also falls in the same complexity category; however, by basing our method on (4.4) rather than (4.2) we reduce the integral dimensions. If we had based our algorithm on (4.2) at each iteration, in order to find $\mathcal{Q}_n^*$, we would need to solve $\arg\max_{\mathcal{Q}_n} I(X; Z_{1:N})$ instead of (4.5). Consequently, for every $1 \leq n \leq N$, (4.13) would involve an N-dimensional integral. By basing our algorithm on the relaxed problem (4.4) we are reducing the dimension of integrals from being N to n, for each $1 \leq n \leq N$.

Numerical multidimensional integrations based on Fubini's theorem [88] suffer from a complexity growing exponentially with the number of dimensions. In such cases, reducing the integral dimensions from N to n reduces the total complexity from $O(NK^N)$ in existing work to $O(K^N)$ in our algorithm, where K is the complexity of a 1-dimensional integral. This is consistent with our observations when obtaining numerical results for Section 4.6.

In our MI-based algorithm, the computations can be further reduced when the measurements are conditionally independent, which is a common assumption in sensor networks [33, 78–80]. When measurements are conditionally independent $p(Y_1, Y_2, \cdots, Y_n|X)$ is factorized to $p(Y_1|X)p(Y_2|X)\cdots p(Y_n|X)$. Therefore, the n-dimensional integral in (13) and (14) $\int_{Y_{1:n}}$, breaks into n 1-dimensional integrals $\int_{Y_1} \cdots \int_{Y_n}$, significantly reducing the complexity of the algorithm from exponential to polynomial.

4.5 Channel-Aware Optimal Quantizers

The discussions up to this point have assumed ideal communication channels between the sensors and the FC. In real distributed sensing systems, due to the non-ideal communication channels the quantized data generated by the sensors might not be received correctly at the FC. This will affect the overall performance of the system. Hence, considering the channel effect in designing the quantizers is crucial [18]. For centralized quantization, [87,89,90] revise the MSE formula to include the channel effect. Then, they jointly optimize the source encoders and the reconstruction levels at the receiver by minimizing this new MSE. For distributed quantization, channel-optimized quantizer design has been developed for hypothesis testing by minimizing the Bayesian cost [91,92]. Recently, the distributed channel-aware quantizer design for multiple correlated sources has been addressed by [93,94], where M source encoders are designed to quantize M correlated sources, in presence of noisy communication channels.

In this section, we design optimal channel-aware quantizers for the distributed quantization of a noisy source using MI measure. To model the communication channel between the sensors and the FC, we assume that the quantizers' outputs are mapped to binary words and communicated to the FC through BSCs. We further assume that the channels are independent. In presence of these noisy channels, we now optimize the quantizers' design by maximizing the MI between the unknown parameter and the channels' outputs. We use the Markov chain property in (4.1) and follow an approach similar to Section 4.4 to solve the optimization problem.

The nth sensor quantizes its analog measurement Y_n to Z_n according to the quantization rule $\mathcal{Q}_n$. The quantizer output value is mapped to its binary representation, making a binary word of size $\log_2 L_n$. Each binary word is sent over a BSC with crossover probability ϵ_n. The received binary word at the channel output is mapped to its decimal representation $T_n \in \mathcal{L}_n$. Due to the channel errors, the received word might not be the same as the transmitted word, hence T_n could be different from Z_n. The channel transition probabilities, i.e., $p(T_n = l|Z_n = k)$; $1 \leq l; k \leq L_n$, are derived based on ϵ_n.

Based on the channel transition probabilities, we can write the MI between X and the received symbols at the channels' output $T_1, T_2, \cdots, T_N$, i.e., $I(X; T_1, T_2, \cdots, T_N)$, or in short $I(X; T_{1:N})$. Then we maximize $I(X; T_{1:N})$ to find the optimal N channel-

aware quantizers. Similar arguments preceding (4.5) are applicable here. Hence, the nth optimal channel-aware quantization rule is obtained as

$$\mathcal{Q}_n^* = \arg\max_{\mathcal{Q}_n} I(X; T_{1:n}). \tag{4.15}$$

By substituting $T_{1:n}$ instead of $Z_{1:n}$ in (4.6) and proceeding similar steps, we get

$$\begin{aligned}\mathcal{Q}_n^* &= \arg\max_{\mathcal{Q}_n} \mathcal{I}(X; T_{1:n}) \\ &= \arg\max_{\mathcal{Q}_n} \int_X \int_{Y_{1:n}} \sum_{T_{1:n}} p(X, Y_{1:n}) p(T_{1:n}|Y_{1:n}) \log p(X|T_{1:n}).\end{aligned} \tag{4.16}$$

Since $Y_{1:n}$, $Z_{1:n}$, and $T_{1:n}$ form a Markov chain $Y_{1:n} \to Z_{1:n} \to T_{1:n}$, we have $p(T_{1:n}|Y_{1:n}) = \sum_{Z_{1:n}} p(T_{1:n}|Z_{1:n}) p(Z_{1:n}|Y_{1:n})$, and therefore

$$\begin{aligned}\mathcal{Q}_n^* = \arg \max_{p(Z_n|Y_n)} \int_X \int_{Y_{1:n}} \sum_{Z_{1:n}} \sum_{T_{1:n}} p(X, Y_{1:n}) p(Z_{1:n}|Y_{1:n}) \\ \times p(T_{1:n}|Z_{1:n}) \log p(X|T_{1:n}).\end{aligned} \tag{4.17}$$

Assuming that the channel between each sensor and the FC is independent from the other channels, $p(T_{1:n}|Z_{1:n})$ can be obtained as $\prod_{m=1}^{n} p(T_m|Z_m)$, where each term is the transition probability of the corresponding sensor-to-FC channel. Following the same steps as in Subsection 4.4.1, we can derive similar results. Note that the term $p(X|T_{1:n})$ in (4.17) depends on the quantization rule $p(Z_n|Y_n)$. First we write

$$\mathcal{I}_{ch}^* = \max_p \max_f \mathcal{I}_{ch}(p, f), \tag{4.18}$$

where p is a short term for $p(Z_n|Y_n)$ and f is a short term for $f(X, T_{1:n})$, such that for all realizations of $T_{1:n} \in \mathcal{L}_1 \times \mathcal{L}_2 \cdots \times \mathcal{L}_n$, $\int_X f(X, T_{1:n}) = 1$. And,

$$\begin{aligned}\mathcal{I}_{ch}(p, f) = \int_X \int_{Y_{1:n}} \sum_{Z_{1:n}} \sum_{T_{1:n}} p(X, Y_{1:n}) \times \\ p(Z_{1:n-1}|Y_{1:n-1}) p(T_{1:n}|Z_{1:n}) p(Z_n|Y_n) \log f(X, T_{1:n}).\end{aligned} \tag{4.19}$$

Then, for a fixed p, $\mathcal{I}_{ch}(p, f)$ is maximized by

$$f^*(X, T_{1:n}) = \frac{\int_{Y_{1:n}} \sum_{Z_{1:n}} p(X, Y_{1:n}) p(Z_{1:n}|Y_{1:n}) p(T_{1:n}|Z_{1:n})}{\int_X \int_{Y_{1:n}} \sum_{Z_{1:n}} p(X, Y_{1:n}) p(Z_{1:n}|Y_{1:n}) p(T_{1:n}|Z_{1:n})}. \tag{4.20}$$

And, for a fixed f, p^* that maximizes $\mathcal{I}_{ch}(p, f)$ is obtained as

$$\begin{aligned}&\forall y \in \mathbb{R} \quad p(Z_n|Y_n = y) = \delta_{l^* y}, \\ &\text{where} \quad l^* = \arg\max_l \int_X \int_{Y_{1:n-1}} \sum_{Z_{1:n-1}} \sum_{T_{1:n}} p(X, Y_{1:n-1}, y) \\ &\qquad \times p(Z_{1:n-1}|Y_{1:n-1}) p(T_{1:n-1}|Z_{1:n-1}) p(T_n|Z_n = l) \log f(X, T_{1:n}),\end{aligned} \tag{4.21}$$

where $l \in \mathcal{L}_n = \{1, 2, \cdots, L_n\}$. Finally, a similar iterative solution as Algorithm 7 can be proposed for finding N channel-aware quantization rules.

4.6 Simulation Results

In this section the performance of our proposed algorithm is demonstrated and compared with other methods using computer simulations. In particular, we examine the performance of our MI-based quantization design for the estimation applications and detection applications, in Sections 4.6.1 and 4.6.2, respectively. The effect of non-ideal channels on the optimal quantization rules is investigated in Section 4.6.3.

4.6.1 Estimation Application

For a distributed sensing system with estimation purposes, the quantized values are used in the FC to estimate the unknown. To compare with [28], where the quantization rules are obtained by minimizing the MSE, we use a similar simulation scenario. The unknown parameter X is distributed according to $\mathcal{N}(0,1)$. Two sensors are involved, i.e., $N = 2$. The measurement noises W_1 and W_2 are additive Gaussian noises with correlation ρ and marginal distribution $\mathcal{N}(0,1)$. The number of quantization levels for both sensors is L. At the FC we use the MMSE estimator to estimate X from the quantized measurements Z_1 and Z_2. Similar to [28], the initial quantization breakpoints are chosen from the optimal quantization rules of Lloy-Max algorithm [22], $\mathit{\Gamma}_1^0 = \mathit{\Gamma}_2^0 = \{-0.982, 0, 0.982\}$.

Our algorithm finds the optimal quantizers $\mathcal{Q}_1^*$ and $\mathcal{Q}_2^*$ by maximizing the MI $I(X; Z_1, Z_2)$. According to (4.4), $I(X; Z_1, Z_2)$ can be broken down as $I(X; Z_1) + I(X; Z_2|Z_1)$. Based on Algorithm 7, first $I(X; Z_1)$ is maximized to find $\mathcal{Q}_1^*$, and consequently $I(X; Z_2|Z_1)$ is maximized to find $\mathcal{Q}_2^*$. Due to this breaking down of the task, the MI $I(X; Z_1, Z_2)$, which is the sum of the two components is maximized in two steps. Fig. 4.2 shows the value of MI at each iteration of the algorithm, for $L = 4$ and $\rho = 0$.

At each iteration of the algorithm, the current quantization rules are used to quantize the measurements Y_1 and Y_2 to Z_1 and Z_2, respectively. These values are then used to estimate X using the MMSE estimator, i.e., $\hat{X} = E\{X|Z_1, Z_2\}$. The estimation performance at each iteration is computed in terms of MSE and is shown in Fig. 4.3. It can be seen from Fig. 4.2 and Fig. 4.3 that by increasing the MI, the MSE of estimation is decreased.

The optimal quantization rules at the end of iterations are represented by the set of breakpoints as $\mathit{\Gamma}_1$ and $\mathit{\Gamma}_2$ in Table 4.1. For different simulation scenarios the final

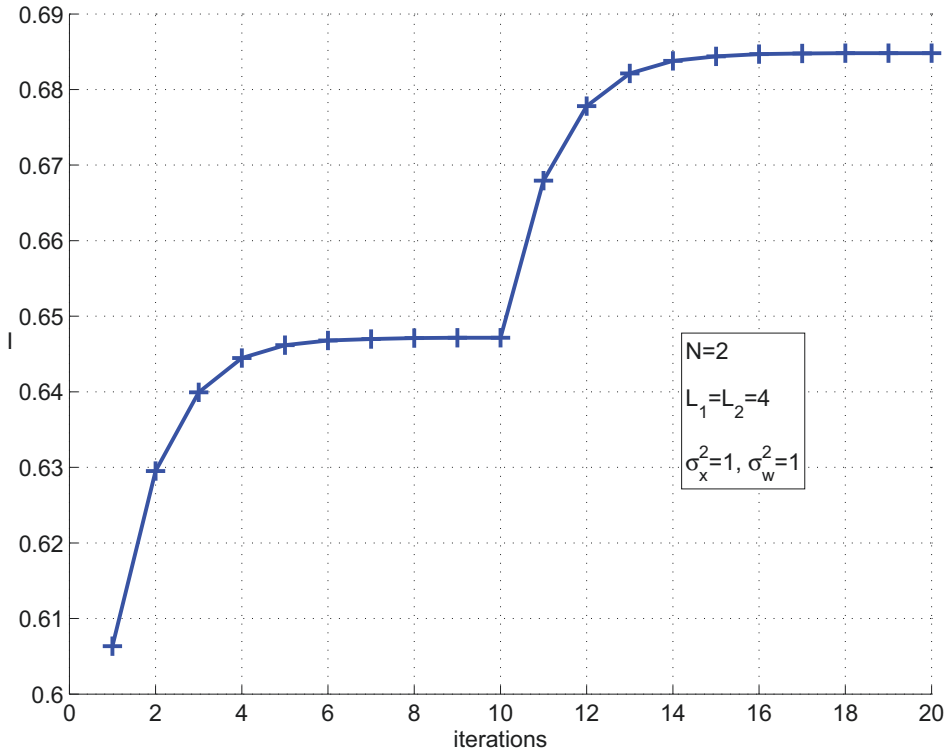

Figure 4.2: Designing the quantization rules by maximizing the MI.

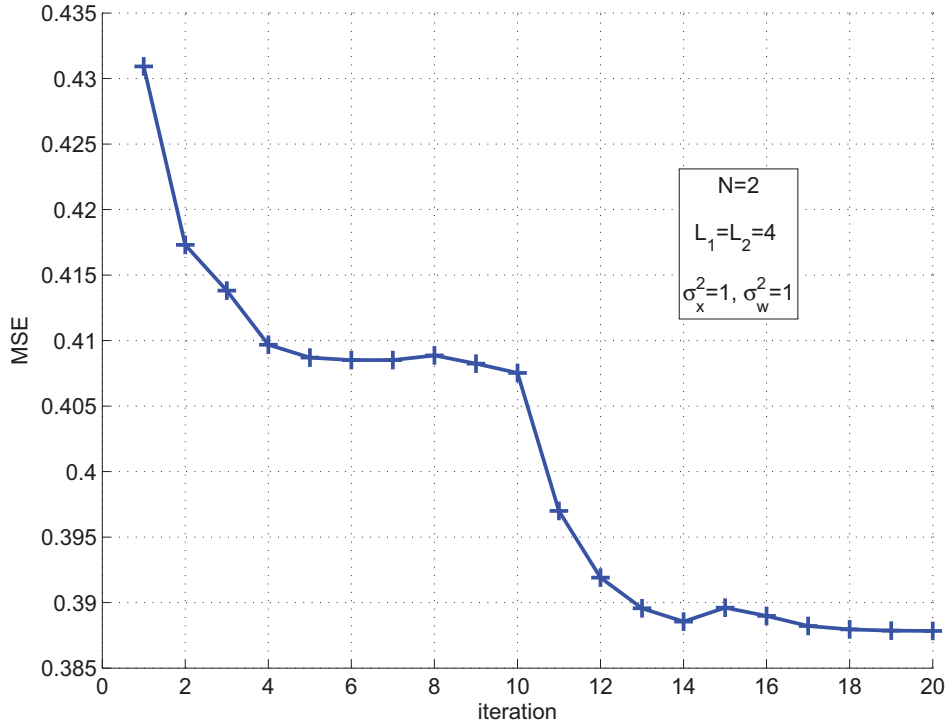

Figure 4.3: MSE change at each iteration.

$\rho = 0$, $L = 4$

method	Optimal Quantization Rules	MSE
MI	$\Gamma_1 = \Gamma_2 = \{-1.37, 0, 1.37\}$	0.3879
Lam	$\Gamma_1 = \Gamma_2 = \{-1.39, 0, 1.39\}$	0.3880

$\rho = 0$, $L = 8$

method	Optimal Quantization Rules	MSE
MI	$\Gamma_1 = \Gamma_2 = \{-2.49, -1.51, -0.73, 0, 0.73, 1.51, 2.49\}$	0.3484
Lam	$\Gamma_1 = \Gamma_2 = \{-2.48, -1.48, -0.71, 0, 0.71, 1.48, 2.48\}$	0.3487

$\rho = 0.85$, $L = 4$

method	Optimal Quantization Rules	MSE
MI	$\Gamma_1 = \{-1.15, 0, 1.15\}$, $\Gamma_2 = \{-2.10, 0, 2.10\}$	0.5170
Lam	$\Gamma_1 = \{-1.13, 0, 1.13\}$, $\Gamma_2 = \{-2.08, 0, 2.08\}$	0.5171

Table 4.1: MSE for estimation, $\sigma_1^2 = \sigma_2^2 = 1$.

quantization rules and the final MSE are shown and compared with the results of Lam algorithm [28]. Comparing with [28], the final quantization rules are different, but, the MSE performances are essentially the same.

We have also implemented the MI-based algorithm using the primal problem in (4.2), and compared the performance results with the implementation based on (4.4). For the three simulation scenarios in Table 4.1, the MSE results for the quantizers of the algorithm based on (4.2) is 0.3878, 0.3482, and 0.5170, respectively. Comparing with the MSE results shown in Table 4.1 it can be seen that the effect of relaxation of the suboptimal problem (4.4) in the performance is negligible.

To investigate a scenario with non-identical measurement noise variances, we have run a simulation where $\sigma_1^2 = 1$, $\sigma_2^2 = 0.5$, and $\rho = 0$. The resulting optimum quantizers are shown in Table 4.2 for $L = 4$ and 8, and are compared with the Lloyd-Max results [22]. It can be seen that the performance of the jointly designed quantizers through our algorithm is better than that of the independent optimal quantizers obtained by the Lloyd-Max algorithm.

For larger N, when the measurements are conditionally independent, i.e., $\rho = 0$, the complexity is feasible. An example is shown in Table 4.3 for $N = 9$, $\sigma_n^2 = 1$; $1 \leq n \leq N$, and $L = 4$, and is compared to Lloyd-Max [22] results.

It is worth mentioning that the performance of our quantization design algorithm improves by increasing the number of quantization levels, as any other quantization problem. This can be investigated by comparing the MSE results of $L = 4$ with

$L = 4$

method	Optimal Quantization Rules	MSE
MI	$\Gamma_1 = \{-1.380, 0, 1.380\}$ $\Gamma_2 = \{-1.135, 0, 1.135\}$	0.310
Lloyd	$\Gamma_1 = \{-0.982, 0, 0.982\}$ $\Gamma_2 = \{-0.694, 0, 0.694\}$	0.332

$L = 8$

method	Optimal Quantization Rules	MSE
MI	$\Gamma_1 = \{-2.43, -1.45, -0.69, 0, 0.69, 1.45, 2.43\}$ $\Gamma_2 = \{-2.05, -1.22, -0.58, 0, 0.58, 1.22, 2.05\}$	0.264
Lloyd	$\Gamma_1 = \{-1.75, -1.05, -0.5, 0, 0.5, 1.05, 1.75\}$ $\Gamma_2 = \{-1.24, -0.74, -0.35, 0, 0.35, 0.74, 1.24\}$	0.281

Table 4.2: MSE for estimation, $\sigma_1^2 = 1$, $\sigma_2^2 = 0.5$.

method	Optimal Quantization Rules	MSE
MI	$\Gamma_n = \{-1.275, 0, 1.275\}$	0.1233
Lloyd	$\Gamma_n = \{-0.982, 0, 0.982\}$	0.1295

Table 4.3: MSE for estimation: $N = 9$.

$L = 8$ (see Table 4.2 and 4.1). However, in noisy observation scenarios the MSE performance is also constrained by the SNR. Therefore, as the number of quantization levels keeps increasing the MSE does not keep dropping with the same slope.

4.6.2 Detection Application

In a distributed sensing system with detection purposes, the FC uses the quantized data to perform a hypothesis testing. We use our method of maximizing the MI to find the optimal quantization rules for the detection scenario and compare the performance with that of Poor algorithm [81], where Ali-Silvey distances [95] are used as the optimization criterion.

To simulate the detection scenario the unknown X can be considered a Bernoulli random variable, which represents the absence (H_0) or presence (H_1) of the signal θ, i.e., $P(X = \theta) = P_\theta$ and $P(X = 0) = 1 - P_\theta$. Each sensor makes an observation of X in additive Gaussian noise and sends the quantized observation to the FC. We use the algorithm in Subsection 4.4.1 to design the optimal rules for quantizing the measurements. Note that since X takes its values from the finite set $\{\theta, 0\}$, the integral over X in all equations translates into a summation over this set. At the FC, the Neyman-Pearson method is used to test the hypotheses.

		$\theta = 2$	$\theta = 4$
$N = 5$	MI:	1.7×10^{-2}	1.0×10^{-5}
	Matsusita:	1.7×10^{-2}	1.1×10^{-5}
	J-divergence:	1.7×10^{-2}	2.1×10^{-5}

Table 4.4: Probability of error for signal detection.

To compare with Poor [81], we assume equally likely H_0 and H_1, i.e., $P_\theta = 1/2$. Also, each sensor quantizes its observation to $L = 4$ levels. And, the measurement noises are i.i.d. with pdf $\mathcal{N}(0, 1)$. The probability of detection error is shown in Table 4.4 for two different signal energies. Our method is indicated by "MI" in the table. The results based on Matsusita distance and J-divergence criteria from [81] are also indicated in the table. It can be seen from the error probabilities that the detection performance of the quantizers designed based on the MI is similar and in some cases better than that found using [81].

4.6.3 Channel Effect

The presence of a non-ideal communication channel between each sensor and the FC affects the design of optimal local quantizers for each sensor. Using the design algorithm developed in Section 4.5, we find the channel-aware local quantizers. The simulation results confirm that the optimal quantizers assuming ideal channels are different from the optimal quantizers in the presence of non-ideal channels.

To compare the channel-aware and channel-unaware quantization schemes we consider an estimation application. The simulation scenario is similar to Subsection 4.6.1. As stated in Section 4.5, we assume that sensors' quantized data are mapped to binary words and sent over a BSC. In the simulation examples we assume that the crossover probabilities are the same for all channels, i.e., $\epsilon_1 = \epsilon_2 = \epsilon$.

Figs. 4.4 and 4.5 show maximization of the MI and minimization of the MSE during the iterations for different values of ϵ, respectively. The final quantizers are given in Table 4.5. It can be seen from Table 4.5 that the optimal quantization solution changes depending on the channel error probability. Consequently, if for instance, one deploys the quantizers designed for $\epsilon = 0$ in a scenario where $\epsilon = 0.05$, the MSE will be 0.497, while using the optimal quantizers designed for $\epsilon = 0.05$, the MSE is 0.485.

$\rho = 0,\ L = 4$		
$\epsilon = 0$	$\Gamma_1 = \Gamma_2 = \{-1.370, 0, 1.370\}$	MSE $= 0.388$
$\epsilon = 0.001$	$\Gamma_1 = \Gamma_2 = \{-1.358, 0, 1.358\}$	MSE $= 0.391$
$\epsilon = 0.01$	$\Gamma_1 = \Gamma_2 = \{-1.260, 0, 1.260\}$	MSE $= 0.411$
$\epsilon = 0.05$	$\Gamma_1 = \Gamma_2 = \{-0.973, 0, 0.973\}$	MSE $= 0.485$

Table 4.5: Optimal quantizers in presence of non-ideal channel.

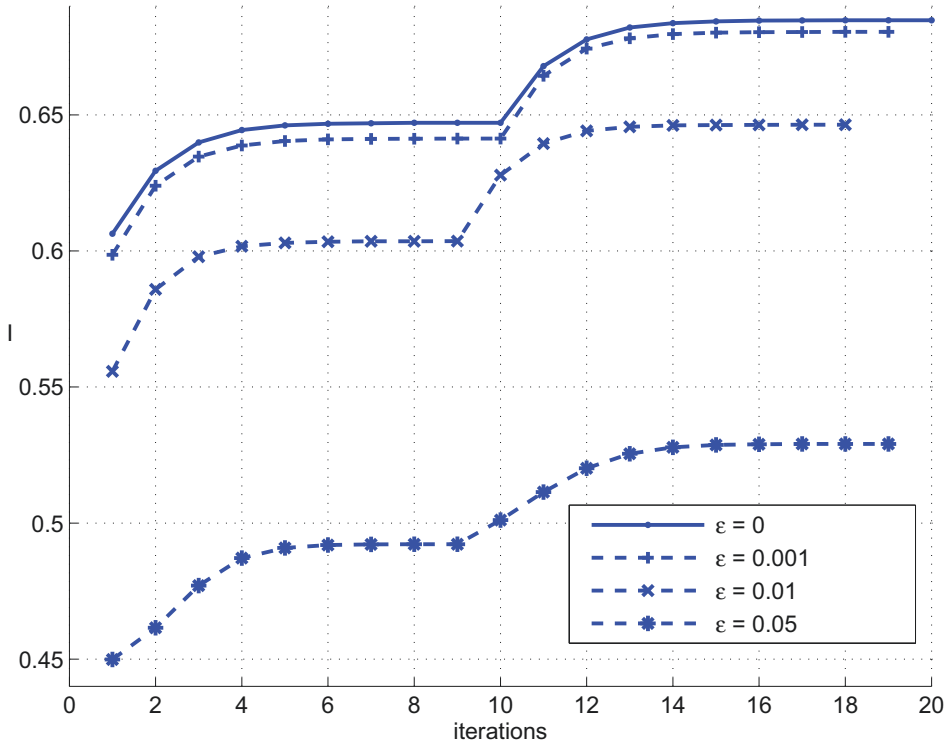

Figure 4.4: Maximizing the MI in presence of non-ideal communication channels

4.7 Conclusion

In this chapter, we proposed an algorithm based on maximizing the MI measure for jointly designing optimal channel-aware local quantization rules for a distributed sensing system. The MI allows us to design general purpose quantizers that later can be deployed for different applications, e.g., estimation or detection. We have shown that the performance of the optimal quantizers based on the MI is essentially the same as the performance of optimal quantizers designed based on other methods that specifically target the estimation or detection application. We also observed that the optimal local quantizers in the presence of non-ideal channels are different from the local quantizers that are optimized without considering the channel

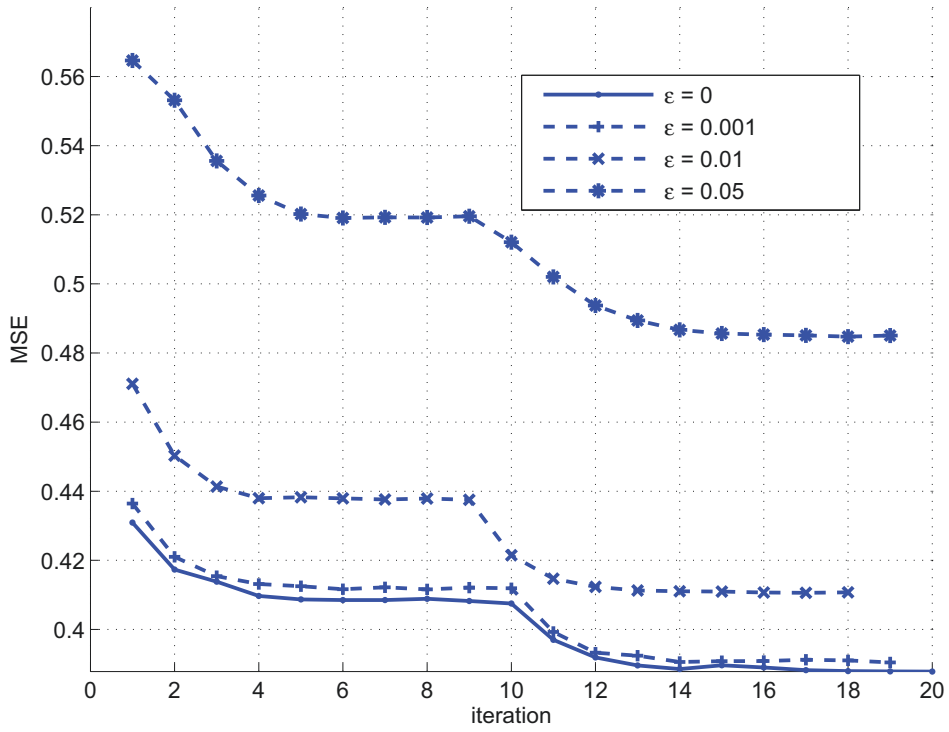

Figure 4.5: MSE of estimation at each iteration of the algorithm.

effect. Our proposed MI-based quantizer design algorithm has significantly lower complexity than other optimal design methods, specifically when the distributed observations are conditionally independent. It can be used in various applications and can integrate channel effects into the design. The studies in this chapter and the related simulations and results are presented in [96].

Chapter 5

Binary Distributed Quantization

In distributed (decentralized) estimation, an unknown must be estimated from its noisy measurements collected at different locations. Due to limited data communication resources, these measurements are typically quantized to be sent to a fusion center, which calculates the estimation of the unknown. In the most stringent condition, each measurement is converted to a single bit for transmission. In this study, we propose a method for generating binary data from analog measurements by introducing some functions called "local binary quantization patterns" (LBQPs). The LBQPs are initially designed so that together they mimic the functionality of a multi-level centralized quantizer. We also improve the design to include some error-correcting capability, increasing the estimation accuracy. The distributed quantization formed by such LBQPs along with the proper estimator proposed in this chapter achieve better estimation performance compared to the existing distributed binary quantization methods.

5.1 Introduction

In this chapter we focus on estimating a random scalar from several noisy observations which have been compressed into one bit. Suppose N noisy measurements of a random parameter X have the form $Y_n = X + w_n$; $1 \leq n \leq N$, where w_n are independent zero-mean measurement noises. If all the measurements are available in the estimator without any distortion, an unbiased estimation with error variance as low as $(\sum 1/\sigma_n^2)^{-1}$ can be achieved, where σ_n is the variance of the measurement noise for the nth observation. When the measurement device(s) and the process-

ing device are not in the same location, the measurements should be transmitted to the FC. Generally, the observations are quantized before transmission using a local quantization rule. In distributed estimation using quantized data, the goal is to design these local quantization rules, as well as the procedure to be used for estimating X from the quantized data. The fundamentals of quantization theory are described in Section 2.2 and an optimum distributed quantization design was proposed in Chapter 4. In this chapter we address the distributed quantization and estimation under very stringent capacity constraints.

Some existing distributed estimation algorithms are based on applying a uniform quantizer to quantize each analog measurement into a few bits [13, 15, 35, 36, 97, 98]. In all these algorithms, the number of quantization bits for each analog measurement is decided based on the SNR. For example in [15], the number of quantization bits for the nth measurement, $1 \leq n \leq N$, is $\log_2(1+V/\sigma_n)$, when X is bounded to $[-V, V]$. In some applications such as WSNs, this means the sensors with higher SNR, i.e., smaller σ_n, will have to send more bits; hence, they consume more power for data transmission. Consequently, the better sensors become more exhausted and die more quickly, which in turn reduces the long-term performance of the estimation task. Another technique has been proposed in [37] which is based on adding some deterministic or random control input to the observation data, prior to the quantization. Therefore, the quantized value of the nth observation is $\tilde{y}_n = \mathcal{Q}_n(Y_n + v_n)$, where v_n is a deterministic or random signal. To find the optimal control input a metric based on the CRLB or equivalently the Fisher information is optimized [37, 99, 100].

Under severely stringent capacity constraints it is preferred to compress/quantize each measurement into only one bit. Distributed estimation based on local binary quantizations has been studied in [14, 34, 36, 78, 101, 102]. In [36], the local binary quantization is performed by comparing the analog measurement value to a fixed threshold in the middle of the analog data range. The estimation performance of a set of local binary quantizers are studied in [102] for asymptotic condition, i.e., $N \to \infty$. The local binary quantizers with a single fixed threshold are not very efficient for high-SNR regime, especially when the number of measurements N is small. To improve the performance, [101] and [14] suggest adaptive thresholds, which are sequentially adjusted according to the previously generated bits. In their approach, the threshold used for quantizing the ith measurement value is adjusted based on the previous $(i-1)$ bits. In [14], to find the unknown parameter X a ML

estimator must be derived every time based on the N generated bits and the N adaptive thresholds.

The estimation based on local binary-quantized observations is further studied in [78], where they show that the estimator performance improves when the local quantizers' threshold gets close to the actual value of the unknown parameter. A way to benefit from this fact is to consider different thresholds τ_k on the analog range to assure that there will always be a threshold close to the true parameter X. Out of the N observations, N_k measurements are quantized according to threshold τ_k, where $\sum_k N_k = N$. Based on the CRLB, they find the set of optimal threshold values τ_k^* and their associated frequencies N_k^*.

In the above distributed estimation methods based on binary data, the generation of each bit is performed using a *single threshold*. However, more advanced local binary quantization schemes can be devised, improving the overall estimation accuracy in the FC. This idea has been practiced by [34]. Based on the distributed estimation method in [34], each measurement is used to estimate one of the bits in the binary representation of the unknown parameter. According to [34], if N measurements are available, $N/2$ of them estimate the first bit of the unknown parameter, $N/4$ estimate the second bit, and so on. So, the bth bit in the binary representation of the unknown parameter is estimated $N/2^b$ times. Hence, the value for that bit is determined by taking the average among the $N/2^b$ binary values, and consequently, X is estimated by combining the final value of the individually estimated bits.

In this work we propose a new method for distributed estimation based on binary-quantized measurements. To do that, we introduce an efficient distributed quantizer to compress each local binary measurement to a bit, and suggest a centralized estimator to infer the unknown from those bits. Therefore, our goal in this chapter is to (i) first formulate the distributed quantization as a set of N LBQPs, which together imitate an L-level quantization, (ii) jointly design these local binary quantizers to find the optimal set of N LBQPs, which can maximize the estimation accuracy of X, and (iii) find a centralized algorithm that combines these binary quantized data to form an accurate estimate of X.

The N binary data generated using our proposed local binary quantizers collectively represent the unknown parameter more accurately compared to the previous binary quantization methods presented in [14,34,36,78], hence achieve a better MSE

performance at the FC. Also, unlike the iterative methods in [26–28], our design of the local quantization rules is globally optimal.

The rest of this chapter is organized as follows. In Section 5.2, the detailed setup of the problem and the required definitions and assumptions are provided. In Section 5.3, the design of a distributed quantizer based on different binary numeral systems is introduced. Section 5.4 proposes optimal LBQPs to improve the estimation performance. In Section 5.5, the appropriate decoder/estimator to be used in the FC is formulated. Finally, in Section 5.7, the simulation results are shown for performance evaluation.

5.2 Problem Setup

Suppose X is a random scalar distributed according to the pdf $p(X)$. A number of noisy measurements of X are observed as

$$Y_n = X + w_n \qquad 1 \leq n \leq N, \tag{5.1}$$

where w_n for $1 \leq n \leq N$ are i.i.d. additive noise. In this chapter, we assume that $p(X)$ is a uniform distribution in the interval $[-V, V]$, and the measurement noise is Gaussian with zero mean and variance σ^2. It is straightforward to modify the proposed distributed estimation method to work with other signal and noise pdfs. An example for Gaussian X is discussed in Section 5.4.3.

Considering the most stringent scenario, only single-bit data transmission is allowed for sending each measurement to the FC. Therefore, each measurement Y_n is quantized to a bit z_n, according to a local binary quantization pattern $\mathcal{P}_n(\cdot)$. Please note that in Chapter 4 the term $\mathcal{Q}$ describes a general quantization rule, whereas in this chapter we have chosen a new term, i.e., $\mathcal{P}$, to emphasize the *binary* quantization rule.

Note that in this study, we do not consider any noise or error added by the channel, i.e., the communication channel is assumed to be error-free. Therefore, all errors discussed here are due to the measurement or quantization noise, not the channel noise. Thus, the goal is to design a set of local binary quantization rules $\mathcal{P}_n(\cdot)$ to be used for quantizing the observations Y_n and also, to design an estimation algorithm that combines the quantized binary data z_n to form an accurate estimate of X at the FC, i.e., $\hat{X}$ (see Fig. 5.1).

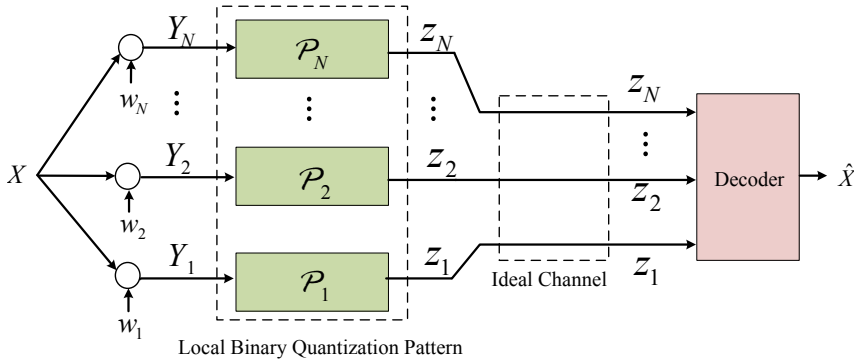

Figure 5.1: The overall distributed estimation method.

Assume that the parameter range $[-V, V]$ is "partitioned" into L "divisions" δ^l; $1 \leq l \leq L$. Compressing the real-valued measurement Y_n to a bit z_n can be performed through introducing a local binary quantization pattern $\mathcal{P}_n$, which is a function mapping each division δ^l; $1 \leq l \leq L$ to a binary value, i.e., 0 or 1. This, quantization of each measurement can be described as

$$z_n = \mathcal{P}_n(l) \quad \text{iff} \quad Y_n \in \delta^l. \tag{5.2}$$

Each LBQP (sometimes just called "pattern" for brevity in this chapter) is an ordered sequence of L binary values, i.e., a binary vector of length L. Wherever the binary value alters between two successive divisions the threshold between those divisions is regarded as an edge in the pattern, see Fig. 5.2. Therefore, a LBQP $\mathcal{P}_n$ is associated with a set of edges, i.e., $\{e_n^1, e_n^2, \cdots, e_n^{\epsilon_n}\}$, where ϵ_n is the number of edges in $\mathcal{P}_n$. These edges define a new partitioning of $[-V, V]$ into $\epsilon_n + 1$ "cells", see Fig. 5.2.

It must be mentioned that, due to the additive Gaussian noise, the analog measurements Y_n are in the range $(-\infty, \infty)$. However, since the desired parameter which must be estimated in the FC is within the range $[-V, V]$, the quantization patterns are defined over this range. Therefore, before quantization, if Y_n is in $(-\infty, -V)$ or (V, ∞) it will be mapped to V or $-V$, respectively.

5.3 Binary Distributed Quantization

In this section we describe how a set of local binary quantization patterns are designed. For now, assume that we have $N = B$ noiseless observations of X. In other words, each observation equals X, which must be quantized into one bit, z_b;

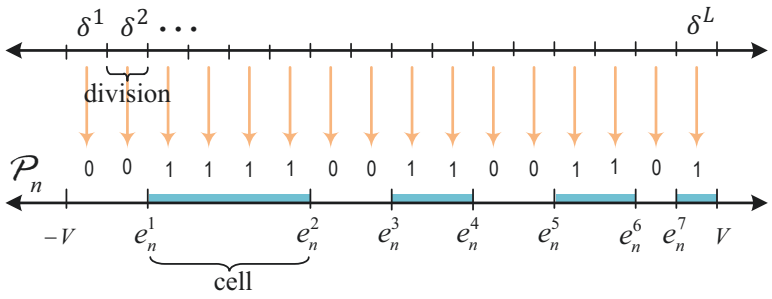

Figure 5.2: A sample local binary quantization pattern

$1 \leq b \leq B$. Since the combinations of B bits can specify maximum of $L = 2^B$ values, one can generate z_b; $1 \leq b \leq B$, so that together they identify the division of X among L distinct divisions within the range $[-V, V]$. In order to achieve that, B LBQPs must be appropriately designed.

Consider the analog range $[-V, V]$ to be partitioned into L equal-length divisions[1], δ^l; $1 \leq l \leq L$. To identify to which division among the total L divisions X belongs, B bits are provided for the FC, using B LBQPs, i.e., $\mathcal{P}_b, 1 \leq b \leq B$. These patterns can be designed so that for each division on the range $[-V, V]$, the B bits together make the B-digit binary label/word assigned to that division. Fig. 5.3 illustrates $B = 4$ LBQPs that together assign 4-digit binary words to the 16 divisions. This assignment is conducted using the "natural" binary numeral system. In other words, reading vertically from left to right, the first division is assigned 0000, the second is assigned 0001, the third is assigned 0010, and so on. If a different binary numeral systems, referred to as the "labeling" scheme in this thesis, is used for labeling the divisions, a different set of B LBQPs would result. This is why the patterns in Fig. 5.3 are identified as $\mathcal{P}_b^{\mathrm{Nr}}$; $1 \leq b \leq B$, where the superscript Nr stands for the natural labeling. If instead of the natural labeling, Gray labeling is used, a set of LBQPs, i.e., $\mathcal{P}_b^{\mathrm{Gr}}$, will be produced, see Fig. 5.4. Clearly, other labeling schemes can be considered.

Similarly, for the noisy scenario, we can quantize each local noisy measurement Y_b, $1 \leq b \leq B$, using one of the patterns $\mathcal{P}_b$, $1 \leq b \leq B$. Depending on where Y_b falls, $\mathcal{P}_b$ decides the bth bit. Since there are $N = B$ measurements, the bth measurement provides the bth digit of the B-digit binary word, for $1 \leq b \leq B$. At the FC, the B bits are used to remake the B-digit binary word and locate an estimation of X.

[1]Considering equal-length divisions is intuitive when we have a uniform source. For non-uniform sources a non-equal partitioning must be considered, see Section 5.4.3.

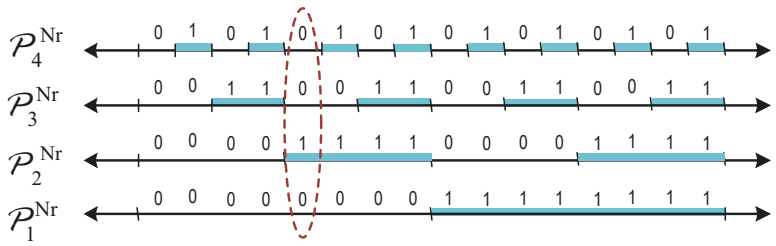

Figure 5.3: The natural basis patterns for $B = 4$.

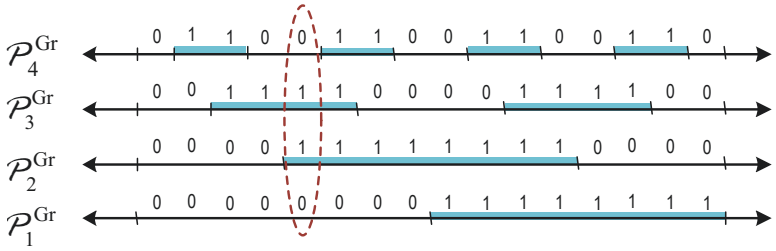

Figure 5.4: The Gray basis patterns for $B = 4$.

The above described method can be viewed as a *distributed quantizer*, since a uniform L-level quantization is implemented using B separate binary quantizers. In a noiseless scenario, the performance of this distributed quantizer is the same as the centralized scalar quantizer, meaning that its MSE is $\Delta^2/12$, where $\Delta = 2V/L$. Moreover, when all observations are equal to X, using either $\mathcal{P}_b^{\mathrm{Nr}}$; $1 \leq b \leq B$, $\mathcal{P}_b^{\mathrm{Gr}}$; $1 \leq b \leq B$, or any other set of B patterns based on different labelings, results in the same MSE, i.e., $\Delta^2/12$. However, in the presence of the measurement noise, any of the B bits of the distributed quantizer might be quantized wrong, leading to a larger estimation error in the FC. In this case, the overall MSE at the FC is affected totally differently by different quantization patterns and also depends on the position of the unknown X. Thus, writing the MSE in closed form will not have a simple or insightful structure. Therefore, in Section 5.6, we derive some performance bounds to compare to the simulation results of our method.

One way to reduce the final MSE is to use better labeling schemes, which results in different LBQPs. In Section 5.4.1, the optimal LBQPs for achieving the best performance are discussed. For uniform X, using the Gray labeling to design the patterns achieves the optimal estimation performance. However, this is not the case for other distributions of X. For Gaussian X, the optimal LBQPs are discussed in Section 5.4.3.

It is possible to further enhance the estimation when more observations are

available. In the same noise level, when $N > B$ measurements are used for a $L = 2^B$-level quantizer, it is possible to estimate the division of X with more reliability. In such cases, the extra measurements can be used to repeat some of the binary digits, and reduce the bit error rate of those bits; i.e., a similar approach as [34]. However, instead of simple bit repetition, one can better employ the extra measurements to get the optimal estimation performance. This is further discussed in Section 5.4.2.

In the following sections, we introduce an algebraic approach to further explain the distributed quantization. Through that, we find the optimal LBQPs to get the best estimation performance when $N = B$ or $N > B$. In particular, for $N > B$, an analogy between error-correcting codes and the distributed quantization is used.

5.4 A Global Optimum Solution

Assume that there are N independent noisy measurements from the unknown X. To achieve the best estimation, one needs to obtain the optimal set of LBQPs to be assigned to the analog measurements. For a fixed B, if $N = B$, the solution to the above problem results in the optimal labeling scheme. If $N > B$, the result is an error-correcting distributed quantization. These two cases are studied separately in the following sections.

5.4.1 Best Labeling Scheme

In Section 5.3, we discussed two sets of LBQPs, resulting from applying two different labeling systems, i.e., natural and Gray. In this section we introduce a technique to obtain many other sets of LBQPs. It is worth mentioning again that if no measurement noise is present all of these sets of LBQPs will have the same performance. However, in the presence of measurement noise, they have different performances in terms of the estimation MSE.

To locate a variable in L distinct divisions (in the interval $[-V, V]$), B bits are needed, where $B = \log_2 L$. To generate such B bits, an eligible set of B local binary quantization patterns can be chosen from a variety of choices. As mentioned in Section 5.2, each pattern is basically a binary vector of length L. Choosing B "eligible local binary quantization patterns" that can identify L different divisions is analogous to choosing B linearly independent binary vectors[2] of size L. The B eligible local binary quantization patterns are called the "basis patterns", for brevity.

[2]Please note that the algebraic calculations are done in the Galois field of 2.

The set of natural quantization patterns in Section 5.3 gives one example set of basis patterns, namely the "natural basis patterns" $\mathcal{P}_b^{\mathrm{Nr}}, 1 \leq b \leq B$, Fig. 5.3. Taking the natural basis patterns as the reference and combining them to generate another set of B linearly independent vectors, we can obtain different set of basis patterns. A $B \times B$ binary matrix of rank B can represent linear combinations of the B reference patterns into a new set of B linearly independent patterns. For example, for $B = 4$ a special combination of natural basis patterns can be shown by

$$G_{BB} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.3}$$

which indicates a combination of the patterns as

$$\begin{aligned} \mathcal{P}_1^{\mathrm{Gr}} &= \mathcal{P}_1^{\mathrm{Nr}} \\ \mathcal{P}_2^{\mathrm{Gr}} &= \mathcal{P}_1^{\mathrm{Nr}} \oplus \mathcal{P}_2^{\mathrm{Nr}} \\ \mathcal{P}_3^{\mathrm{Gr}} &= \mathcal{P}_2^{\mathrm{Nr}} \oplus \mathcal{P}_3^{\mathrm{Nr}} \\ \mathcal{P}_4^{\mathrm{Gr}} &= \mathcal{P}_3^{\mathrm{Nr}} \oplus \mathcal{P}_4^{\mathrm{Nr}}, \end{aligned} \tag{5.4}$$

where $\oplus$ means modulo-2 summation of the L-valued binary vectors representing the patterns. The new set of basis patterns obtained as above are the Gray patterns, Fig. 5.4.

We focus on the natural basis patterns as the original patterns, and their linear combinations, in the form of G_{BB}, for producing new patterns. By searching all matrices of type G_{BB}, one can find the best basis patterns with the lowest MSE, for each SNR level. As mentioned above, G_{BB} must be full rank. Moreover, a permutation of the vectors of G_{BB} results into the same set of patterns. Hence, the size of the search space is

$$\Omega_1(B) = \frac{1}{B!} \prod_{i=1}^{B} \left(2^B - 2^{i-1}\right). \tag{5.5}$$

5.4.2 Error-Correcting Patterns

As mentioned earlier, having $N > B$ measurements available, the estimation accuracy can be improved by repeating some of the local binary quantization patterns or more effectively, by introducing new patterns. In the later approach, N patterns are obtained from B basis patterns, which together generate N-bit words to identify

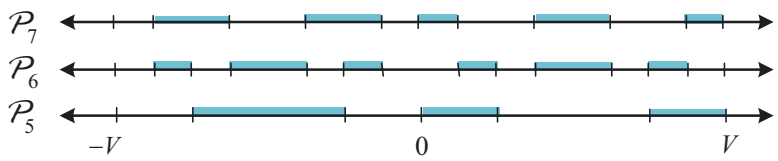

Figure 5.5: LBQPs for the extra three bits.

$L = 2^B$ divisions. Obviously, using N-bit words instead of B-bit words is a redundant way of labeling an L-division quantization. However, this redundancy enables some error correction possibility in estimating the division of X. This property is analogous to error-correcting property in channel coding, where some redundant bits are added to the B-bit messages to make N-bit codewords. This happens by adding extra vectors to the generator matrix [103].

For distributed quantization, among the N patterns used to quantize the N measurements, there must be at least B linearly-independent patterns. Hence, the N patterns are produced by linear combination of a set of B basis patterns according to a $B \times N$ matrix of rank B. For example, consider the matrix

$$G_{BN} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}, \tag{5.6}$$

which is used to produce N patterns from the B natural basis patterns. In this example, the first part of G_{BN} is a 4×4 matrix similar to (5.3). Therefore, there will be the 4 Gray basis patterns among the 7 patterns produced by G_{BN}. Fig. 5.5 shows the 3 extra patterns.

Different G_{BN} matrices can be used to produce different sets of patterns, each having a different estimation performance. A repetition strategy, in which some of the basis patterns are repeated for quantizing the extra measurements, is a special case of the explained approach. However, by searching among different matrices one can find a set of N LBQPs that have better performance than a mere repetition scheme. The optimal set of LBQPs for each values of N and B depends on the SNR, and can be obtained by exhaustive search. The size of the search space is given by

$$\Omega_2(B, N) = \Omega_1(B) \times \binom{2^B - 1 + (N - B - 1)}{N - B}. \tag{5.7}$$

In Section 5.7 some optimal results are shown and discussed. It is worth mentioning that, the search for finding the optimal patterns is done in the FC and

happens only once. After the optimal set of LBQPs is obtained, each pattern is assigned to each observation. As long as the measurement noise variance does not change a lot, the patterns do not need to be updated.

It is worthwhile pointing that there is a difference between the functionality of error-correcting codes when used against channel noise and when, in this case, used in the distributed estimation problem. In channel coding, the source of bit error is the additive channel noise which contaminates the coded bits after they are sent on the channel. On the contrary, in the case of distributed quantization, the bit error happens during the quantization process when the bits are being generated. The main difference this causes is that unlike the channel coding, the a priori bit error probability (BEP) is not the same for all bits. Even in a homogeneous scenario with i.i.d. noise for all the observations, i.e., same σ, the probability of error for each bit depends on its LBQP and how small are the *cell lengths* in that pattern. Moreover, the BEPs also depend on the value of X. Because of these fundamental differences, known optimal channel codes are no longer optimal for distributed estimation. Also, a separate study is required for the decoder/estimator of the proposed error-correcting quantization method.

5.4.3 Gaussian Source

Up to this end we had assumed that the unknown parameter X is uniformly distributed. Therefore, in Section 5.3, we had started our method by first assuming a uniform partitioning of the range $[-V, V]$ into L equal divisions. If the parameter X is not uniformly distributed, the uniform partitioning of the range is not optimal. However, still the same methodology described in Sections 5.4.1 and 5.4.2 can be used to design the optimal patterns, but with a different partitioning of the range of X. As an example, we discuss the Gaussian source. The results of applying our distributed quantization method for Gaussian parameter are presented in Section 5.7.

For a Gaussian source, we assume that the range of X, i.e., $(-\infty, \infty)$, is partitioned according to the principals in the Lloyd-Max algorithm [22,23] for centralized quantization. For example, for $X \sim \mathcal{N}(0, 1)$, the optimal 3-bit centralized quantization has the range of X partitioned into 8 divisions as $\{-\infty, -1.748, -1.05, -0.5006, 0, 0.5006, 1.05, 1.748, \infty\}$. To derive the B LBQPs for a Gaussian noise, we take the same steps as in Sections 5.4.1 and 5.4.2, except that the value of edges for every

pattern is a subset of the above edges.

5.4.4 Suboptimal Search Strategy

Due to the nature of the optimization problem, an exhaustive search must be performed to find the optimal LBQPs. In many practical scenarios, B and N are small quantities, and the search is feasible. For those values of B and N which the search space size is too big, sub-optimal strategies can be suggested to reduce the complexity. Some of those strategies are discussed here.

Strategy 1: When a B-bit precision is practiced with N sensors, there must be always B linearly independent patterns among the N LBQPs. Based on the simulation results, it can be investigated that for uniform X and Gaussian noise, there is always the same B independent patterns for every SNR value. In this scenario, the B quantization patterns derived based on Gray labeling, i.e., $\mathcal{P}_1^{\mathrm{Gr}}, \cdots, \mathcal{P}_B^{\mathrm{Gr}}$, are always among the N optimal quantization patterns. Therefore, for uniform X and Gaussian noise, to design the N patterns for any SNR, one can fix B of them to be the Gray patterns. This reduces the size of the search space from $\Omega_2(B, N)$ to $\Omega_2'(B, N)$, which is equal to the second term of the product in (5.7).

Strategy 2: Another simplifying strategy can be practiced for big values of N. During the simulations, it was observed that when N gets bigger while B is constant, some patterns are repeated among the optimal patterns. Therefore, even for very big N, one can limit the search to a smaller space, e.g., $N_0 < N$, find the optimal patterns for that subspace, and repeat the same N_0 patterns for the rest of the observations.

Both of the above strategies can be combined to reduce the search space. For example, for a uniformly distributed unknown parameter with Gaussian measurement noises, a suboptimal search method is explained as follows. For $1 \leq n \leq B$, one uses $\mathcal{P}_1^{\mathrm{Gr}}, \cdots, \mathcal{P}_B^{\mathrm{Gr}}$. For $B+1 \leq n \leq N_0$, a search is performed on a space with size $\Omega_2'(B, N_0)$. For $N_0+1 \leq n \leq 2N_0$, $2N_0+1 \leq n \leq 3N_0$, $\cdots$, the same N_0 patterns are repeated. A third strategy which further reduces the complexity is presented in Section 5.6.

5.5 Decoding

In the FC, a decoding method must be used to estimate the unknown parameter from the received binary-quantized measurements z_n. Here, we explain two types

of decoders, i.e., the discrete and continuous decoders, which estimate X based on the discrete and continuous a posteriori distributions, respectively. For the sake of better understanding the overall behavior of the distributed estimation method, we first explain the discrete decoder. In both estimators the receiver only needs to do the computations once at the beginning of the algorithm to build a lookup table, which is used during the estimation procedure.

5.5.1 Discrete A Posteriori

The distributed quantization is used to locate X in one of the discrete divisions δ^l; $1 \leq l \leq L$. Therefore, to estimate X at the fusion centre, one can identify the *division* of X from the received bits. In this section we propose a maximum a posteriori (MAP) estimator that decodes the N received bits to estimate the division of X among the L divisions.

As discussed in Section 5.4, for each value of X a set of N LBQPs together generate an N-bit "codeword". Assuming no measurement noise, there are $L = 2^B$ different codewords, i.e., $\boldsymbol{c}^l = [c_1^l c_2^l \cdots c_N^l]$; $1 \leq l \leq L$. There is a one-to-one correspondence between these valid codewords and the L divisions, i.e., δ^l; $1 \leq l \leq L$. In the presence of measurement noise, each of the N bits, i.e., z_n; $1 \leq n \leq N$, might be wrong, resulting in a "received word"[3] $\boldsymbol{z} = [z_1, z_2, ..., z_n]$, which might include some bit errors. The discrete decoder's function is to find the most likely valid codeword based on the received N-bit word $\boldsymbol{z}$, which in turn results into estimating the division of X. This estimator can be reduced to a lookup table that can be saved in the receiver.

To build the lookup table of the discrete estimator, the a posteriori probabilities, of all the L valid codewords conditioned on the received word $\boldsymbol{z}$ are calculated. The codeword with the maximum probability is chosen as the decoder's decision corresponding to each value of $\boldsymbol{z}$. The center of mass [22] for that division is recognized as the estimated value $\hat{X}$. This decoder is therefore referred to as the discrete MAP estimator. The a posteriori probability of each codeword can be written as

$$P(\boldsymbol{c}^l|\boldsymbol{z}) \propto P(c^l) \prod_{n=1}^{N} P(z_n|\boldsymbol{c}^l), \tag{5.8}$$

where $P(c^l)$ is the a priori probability of codeword c^l, which is equal to $P(\Gamma^{l-1} \leq$

[3]Note that the notation "received" in this chapter does not imply a communication channel or channel error.

$X \leq \Gamma^l$), where Γ^{l-1} and Γ^l indicate the left and right edges of the lth quantization division δ^l, corresponding to $\boldsymbol{c}^l$. For uniform X, $\Gamma^{l-1} = -V + (l-1)\Delta$, and $\Gamma^l = -V + l\Delta$.

Since each valid codeword is associated with one of the L divisions δ^l; $1 \leq l \leq L$, each term of the product in (5.8) can be described as

$$P(z_n|\boldsymbol{c}^l) = P(z_n|\Gamma^{l-1} \leq X \leq \Gamma^l) = \frac{\int_{\Gamma^{l-1}}^{\Gamma^l} P(z_n|X)P(X)dX}{\int_{\Gamma^{l-1}}^{\Gamma^l} P(X)dX}. \tag{5.9}$$

The denominator in the second line is by definition $P(c^l)$, the a priori probability of the codeword c^l, which for uniform X is equal for every c^l. Remembering that z_n can be either 0 or 1, (5.9) can be written as

$$P(z_n|\boldsymbol{c}^l) = \int_{\Gamma^{l-1}}^{\Gamma^l} \mathrm{P1}_n(X)^{(z_n)} \left(1 - \mathrm{P1}_n(X)\right)^{(1-z_n)} P(X)dX, \tag{5.10}$$

where $\mathrm{P1}_n(X)$ is the probability that the nth bit is 1 when the parameter value is X, i.e., $p(z_n = 1|X)$. The formula to calculate $\mathrm{P1}_n(X)$ is given in Appendix B.3. Using (5.10) and (5.8) one can find the a posteriori probability of the codewords.

At the beginning of the algorithm a decoding table is formed by calculating the a posteriori probabilities for different values of the received word, i.e., $\boldsymbol{z}^k$; $1 \leq k \leq 2^N$, and different valid codewords $\boldsymbol{c}^l$, $1 \leq l \leq L$. This calculation is done once in the FC and does not need to repeat every time. The decoding table has 2^N entries, one for every possible value of $\boldsymbol{z}$, associating it with a codeword which has the highest a posteriori probability $P(\boldsymbol{c}^l|\boldsymbol{z})$. For every estimation instance, based on the received word, the decoder chooses the corresponding codeword from the table as the most likely codeword and announces the center of mass of the associated division as $\hat{X}$. It is easy to see that he lookup table depends on the patterns set, as well as σ.

The discrete MAP decoder is based on finding the a posteriori probability of the valid codewords, which have one-to-one correspondence with the division of X. Therefore, they can only locate X with a limited precision. To better estimate the analog parameter X, a continuous decoder can be used instead of the discrete decoder, resulting in better estimation performance.

5.5.2 Continuous A Posteriori

The continuous decoder is designed based on the continuous a posteriori pdf of X. Using the a posteriori pdf, a MMSE or a MAP estimator can be designed. The later

is an unbiased estimator, as proved in appendix B.2. Again, similar to the discrete decoder, the receiver only needs to do the computations once at the beginning of the algorithm to build a lookup table, which is used during the estimation procedure.

The continuous a posteriori distribution of X, i.e., $P(X|\boldsymbol{z})$, can be written as

$$\begin{aligned} P(X|\boldsymbol{z}) &\propto P(X) \prod_{n=1}^{N} P(z_n|X) \\ &\propto P(X) \prod_{n=1}^{N} \mathrm{P1}_n(X)^{(z_n)} (1 - \mathrm{P1}_n(X))^{(1-z_n)}. \end{aligned} \tag{5.11}$$

The factorization in (5.11) is valid since the measurement noises as well as the quantization procedures are independent for different measurements. Once the above a posteriori conditional distribution is derived, one can find $\hat{X} = \arg\max_X P(X|\boldsymbol{z})$ or $\hat{X} = E\{X|\boldsymbol{z}\}$ as a MAP or MMSE estimator of X, respectively. In the FC, the following calculations are done once to build a decoding table. For each possible received word $\boldsymbol{z}^k$; $1 \leq k \leq 2^N$, the MAP or MMSE estimation $\hat{X}^k$ is calculated to build an entry in the lookup table, mapping $\boldsymbol{z}^k$ to $\hat{X}^k$. Every time an N-bit word is received at the decoder this lookup table is used to find $\hat{X}$.

It is easy to investigate that the MSE performance of the continuous decoder is better than the discrete one. This is while the computational complexity of the continuous decoder is only slightly higher than the discrete decoder. The performance of different decoders are studied through simulations in Section 5.7, and compared with the CRLB for the optimal unbiased estimator.

5.6 Performance Bounds

Analyzing the MSE performance of the discrete MAP estimator is not straightforward. Even in the case of $N = B$, where the decoder is very simple, the MSE equation derived in [104] is not in closed form. Therefore, we use some bounds as a benchmark to study the performance of our method.

The first bound is based on the rate-distortion theory, which provides a lower bound on the MSE of estimation. The distortion-rate function $D(R)$, determines the minimum distortion, in our problem the minimum MSE, that can be achieved when an analog source is compressed with a rate R. The distortion-rate function for centralized estimation of a Gaussian source from a noisy observation under a rate constraint R is derived in [55], and is used as a lower bound for the distortion

(MSE) in the distributed estimation-quantization problem. For a scalar parameter with Gaussian distribution $\mathcal{N}(0, \sigma_x^2)$, which is observed through additive Gaussian noise with variance σ^2, $D(R)$ is calculated as [55]

$$D(R) = \sigma_x^2 - \frac{\sigma_x^4}{\sigma_x^2 + \sigma^2}(1 - 2^{-2R}). \tag{5.12}$$

This distortion bound has been compared to the performance results of our method for a Gaussian parameter, in Section 5.7.

For the case of uniform parameter, the distortion-rate function of an estimation-quantization system is not well studied in the literature. Here, we have approximated a lower bound for the distortion-rate function. The bound is calculated in Section 5.7 and compared with the results of our distributed estimation method. The details can be seen in Appendix B.1.

To evaluate the performance of our estimator, we have also calculated the CRLB for a set of LBQPs. In our distributed estimation method, for the case of Gaussian or uniform X and Gaussian noise, the CRLB can be the indicator of the best estimation performance if the estimator is unbiased. The proof of unbiasedness of our MMSE estimator is given in Appendix B.2. The CRLB of the estimation method is given by

$$\left(\sum_{n=1}^{N} \int_X \{ \frac{(\frac{\partial \mathrm{P1}_n(X)}{\partial X})^2}{\mathrm{P1}_n(X)} + \frac{(\frac{\partial \mathrm{P1}_n(X)}{\partial X})^2}{1 - \mathrm{P1}_n(X)} \} p(X) dX + J_2 \right)^{-1}, \tag{5.13}$$

where for a uniform X, $J_2 = 0$, and for a Gaussian X with zero mean and variance σ_x^2, $J_2 = 1/\sigma_x^2$. See Appendix B.3 for derivation of (5.13).

The CRLB of three quantization methods are shown in Fig. 5.6, by numerically evaluating the integral in (5.13). The first method is the fixed threshold quantization [36], the second method is the set of LBQPs obtained by natural labeling, and the third is the method based on Gray labeling. It can be seen that in general, the relative performance of the quantization methods depends on the SNR. Therefore, it can be inferred that at each SNR, a set of optimal LBQPs which has the best performance can be found.

strategy 3: Based on the CRLB, a strategy can be devised to reduce the computational complexity of the search method described in Section 5.4. As explained there, the optimal patterns are found by minimizing the MSE. That is to say, for every candidate set of N patterns, the MSE of estimation is found through simulation, and finally the set of patterns with the lowest MSE is selected. To reduce the

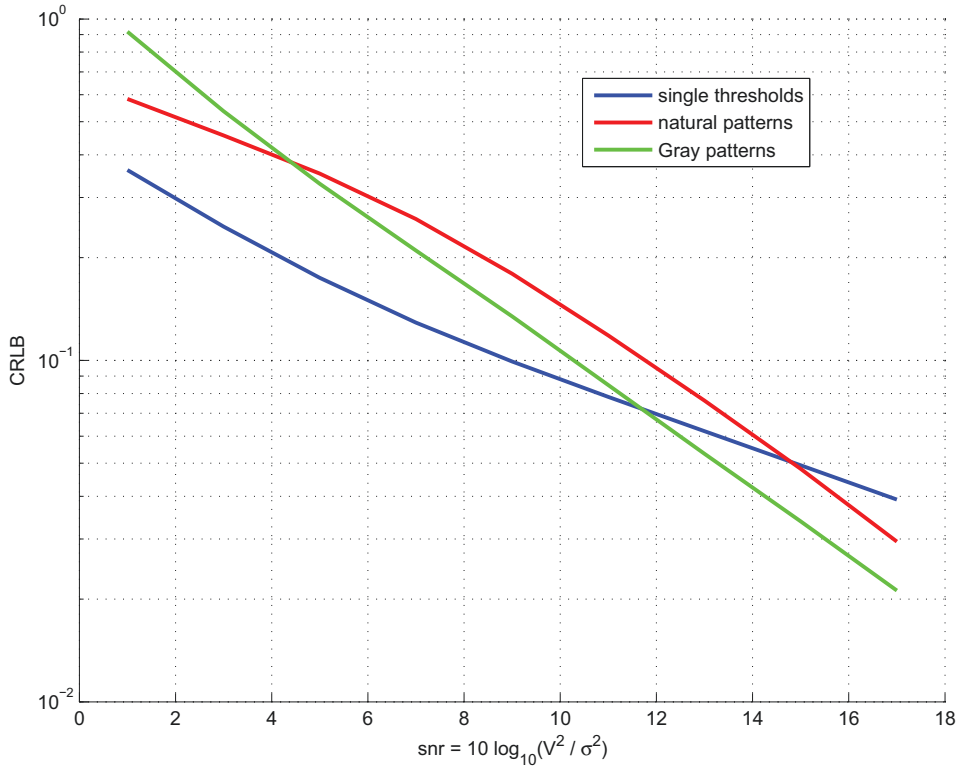

Figure 5.6: CRLB for the different local binary quantization patterns.

complexity, we suggest to use an MSE bound, i.e., the CRLB, instead of finding the actual MSE. So, for every candidate set of LBQPs, the CRLB is calculated based on (5.13), and the set with the lowest CRLB value is selected. Using this strategy along with the other two strategies in Section 5.4.4, the complexity of finding the LBQPs will be reduced greatly. The MSE results for the suboptimal patterns have been compared with the optimal patterns in Section 5.7.

5.7 Numerical Results

In this section, different simulation results are discussed in order to study the performance of the proposed distributed estimation method. Both scenarios in Section 5.4.1, where $N = B$, and Section 5.4.2, where $N > B$, are considered. The results are shown for two distributions of the unknown parameter, i.e., uniform and Gaussian. For the case of uniform unknown parameter, X is uniformly distributed in the interval $[-1, 1]$. For Gaussian X, it is distributed according to $\mathcal{N}(0, 0.1)$. Comparison with other methods, such as [14, 34, 36, 78] are discussed. The performance is evaluated in terms of the MSE. In all simulations, the measurement noise

$\mathcal{P}_1$	0	0	0	0	1	1	1	1
$\mathcal{P}_2$	0	0	1	1	1	1	0	0
$\mathcal{P}_3$	0	1	1	0	0	1	1	0
$\mathcal{P}_4$	0	1	1	0	0	1	1	0
$\mathcal{P}_5$	0	1	1	0	0	1	1	0
$\mathcal{P}_6$	0	0	1	1	0	0	1	1
$\mathcal{P}_7$	0	1	0	1	0	1	0	1
$\mathcal{P}_8$	0	1	0	1	1	0	1	0

Table 5.1: Optimal patterns for uniform X, $B = 3$, $N = 8$, SNR= 18.2dB.

is i.i.d. zero mean Gaussian with variance σ^2. Note that, this SNR is only related to the measurement noise, while the channel is assumed error-free.

Fig. 5.7 shows the performance of the distributed estimation method for a uniform source. Since X is uniform, $E\{X^2\} = V^2/3$; therefore, the average SNR is $10\log_{10}(V^2/3\sigma^2)$. At each SNR, the optimal set of local binary quantization patterns was found by searching all matrices G_{BN} for $B = 3$ and $N = 3$, 5 and 8. For the case of $N = 8$, the results of the suboptimal search method based on both *strategy* 1 and *strategy* 3 are also shown in the figure. Note that, for all cases, the continuous MMSE estimator is used in the decoder. The results show that when $N = B$ the optimal pattern set for every SNR is the Gray basis patterns. When $N > B$, for each SNR a different set of optimal patterns are found, reducing the estimation error. For example, for the case of $B = 3$ and $N = 8$ the optimal patterns for SNR= 18.2dB are presented in Table 5.1. From the table, we can see that the patterns $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}_3$ are the 3 Gray basis patterns, and the rest are linear combinations of those.

For the case of $N = 8$ the performance of our algorithm is compared with the distributed estimation proposed by Luo *et al.* [34]. Based on Luo's algorithm, $N/2 = 4$ measurements are quantized to the first bit in a natural binary system, $N/4 = 2$ measurements are used to estimate the second bit, and the remaining 2 measurements are used for the third bit. As can be seen in Fig. 5.7, the optimal sets of 8 LBQPs for different SNRs outperform the method of [34].

The three different decoding methods based on the discrete and continuous a posteriori functions in Section 5.5 are compared in Fig. 5.8. The first decoding method is the discrete MAP estimation based on the discrete a posteriori probability of the codewords discussed in Section 5.5.1. The other two methods are the continuous MAP and MMSE estimations based on the continuous a posteriori

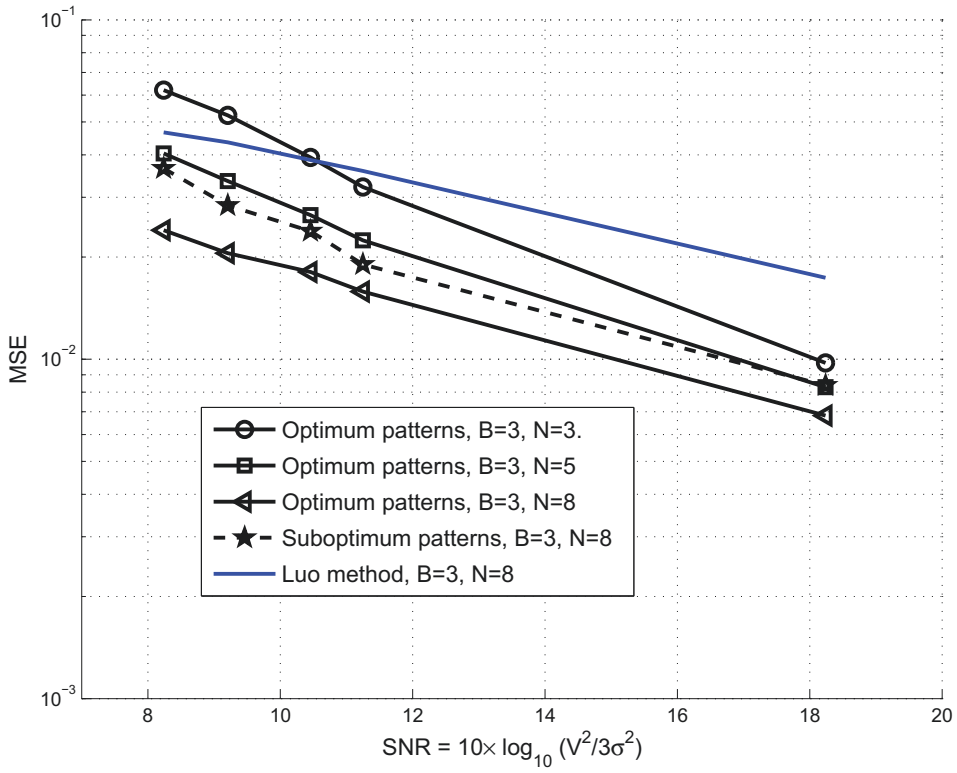

Figure 5.7: Effect of error-correcting patterns in estimation performance.

$\mathcal{P}_1$	0	1	1	0	0	1	1	0
$\mathcal{P}_2$	1	0	0	0	1	1	1	0
$\mathcal{P}_3$	0	1	0	1	1	0	1	0

Table 5.2: Optimal patterns for Gaussian X, $N = B = 3$, SNR= 5.2dB.

density of X in Section 5.5.2. Also, the CRLB and the rate-distortion bound for uniform X (Appendix B.1), have been shown for comparison.

For a Gaussian parameter with zero mean and variance $\sigma_x^2 = 0.1$, the results for $N = B = 3$ are shown in Fig. 5.9. The average SNR in this case is $10\log_{10}(\sigma_x^2/\sigma^2)$. Please note that as explained in Section 5.4.3, the values for division edges of the centralized quantizer, i.e., Γ^l; $1 \leq l \leq L$ are set to $\sigma_x \times \{-1.748, -1.05, -0.5006, 0, 0.5006, 1.05, 1.748\}$ [22]. As an example, the optimal patterns for SNR= 5.2dB are presented in Table 5.2. Note that the patterns are lineally independent, but unlike the case of uniform X, the optimal patterns are not the Gray basis patterns. The MSE results are compared with the rate-distortion bound in (5.12).

Fig. 5.10 compares the performance of our proposed method with other binary quantization methods, for $N = 10$. The circle points in Fig. 5.10 show the MSE for

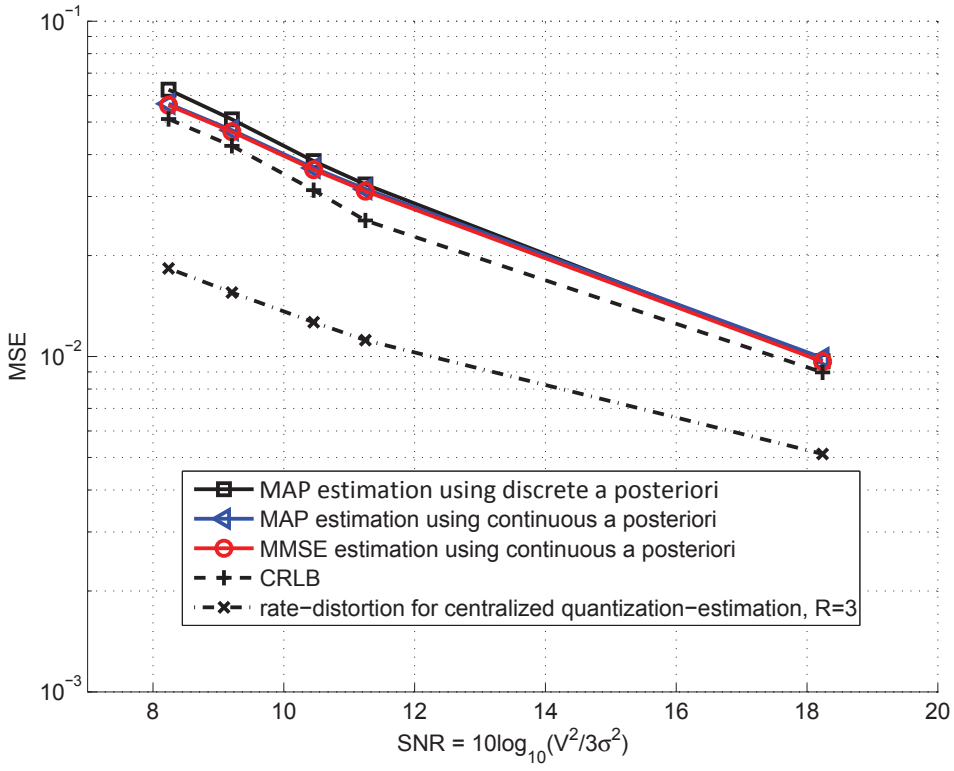

Figure 5.8: Performance of different decoding methods, for Gaussian X.

the suboptimal patterns found using the three suboptimal search strategies. For the method proposed by Ribeiro [78], the CRLB for the optimal solution is indicated. Two adaptive quantization methods proposed by Fang [14], i.e., AQ-VS and AQ-ML are also shown in the figure. For the AQ-ML, before quantizing each measurement a ML estimation must be calculated. It can be seen from Fig. 5.10 that for higher SNRs our method has lower MSE compared to [78] and [14], while its complexity for "each estimation job" (as long as the SNR does not change a lot) is less than those algorithms.

5.8 Conclusion

In this study we proposed a new method for distributed (decentralized) estimation from binary-quantized noisy measurements. The estimation method is based on designing efficient local binary quantization patterns to generate a single bit from each analog measurement and an appropriate decoder to be used at the processing center for estimating the unknown. The results show that for Gaussian X, when B measurements are used to achieve an estimation with B-bit precision, the optimal

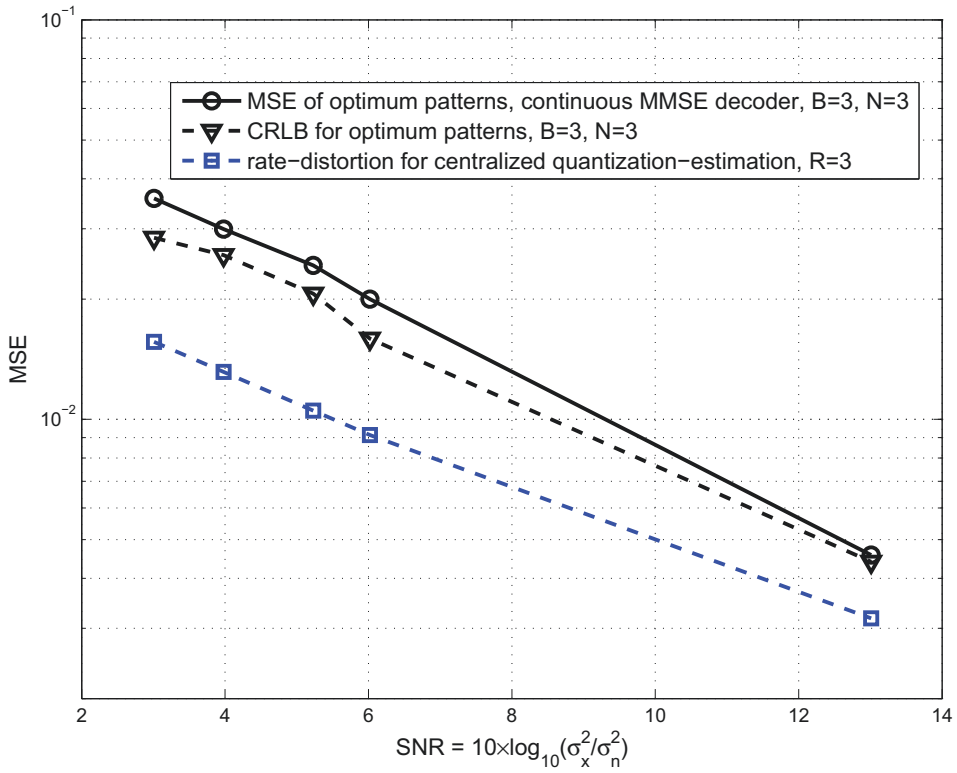

Figure 5.9: Performance of our distributed estimation method for Gaussian parameter with $\sigma_x^2 = 0.1$, $B = 3$.

B patterns are those based on the Gray labeling for all SNRs. For scenarios where $N > B$ measurements are available, extra local binary quantization patterns are designed to achieve an error correcting capability compared to the case when only B measurements are available. We also proposed some suboptimal strategies to reduce the complexity of the search especially for bigger values of N and B. The performance results have been compared with other distributed estimation methods with binary quantization. The studies in this chapter and the related simulations and results are presented in [104, 105].

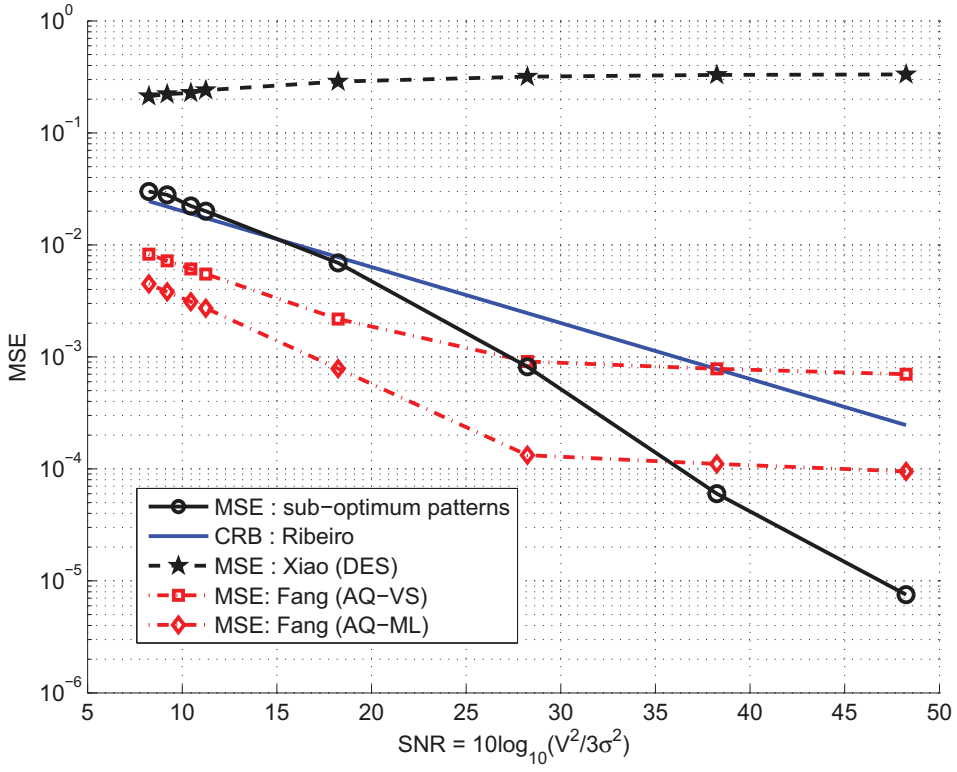

Figure 5.10: Performance comparison for different methods for $N = 10$.

Chapter 6

Optimum Bit-Sensor Assignment

6.1 Introduction

In a network where all sensors have the same measurement qualities, it does not matter which sensor is assigned which local binary quantization pattern, i.e., the bit-sensor assignment does not affect the final estimation performance. However, in many practical WSNs, sensors do not have the same measurement quality (inhomogeneous environments), for instance, due to their different distance to the object of interest. We will discuss that in such conditions the bit-sensor assignment affects the estimation performance. We then propose an algorithm based on Hungarian method [61] to find the best assignment that minimizes the MSE of estimation.

6.2 Problem Definition

Suppose the unknown parameter X is uniformly distributed within the range $[-V, V]$. Each sensor n; $1 \leq n \leq N$ has a noisy measurement of X as $Y_n = X + w_n$, where w_n is the measurement noise of the nth sensor, which has Gaussian distribution with zero mean and variance σ_n^2. The measurement noises of different sensors are assumed to be independent. Each sensor quantizes its measurement to one bit according to its local binary quantization pattern, and sends it to the FC, where an estimate of the parameter is made from N received bits. We assume that sensors' transmitted data is received at the FC with no error. Upon receiving all bits at the FC, a reconstruction point is calculated among the 2^N possible values, and is regarded as the estimation of X, i.e., $\hat{X}$.

When sensors' measurement qualities are not the same, the MSE of estimation

depends on the order in which the N local binary quantization patterns are assigned to the N sensors. In this chapter, we propose a method to efficiently find the optimum bit-sensor assignment that minimizes the MSE among the $N!$ possible assignments. Our proposed algorithm is run at the fusion centre. Having the knowledge of sensors' noise variances, the fusion centre decides a local binary quantization pattern for each sensor. The quantizer information for each sensor is then transmitted to that sensor by the fusion centre. The bit-sensor assignment is such that exactly one quantization rule is assigned to each sensor, and exactly one sensor is assigned for each quantization pattern. Therefore, $\mathcal{P}_b$, $1 \leq b \leq N$, is assigned to sensor n_b, where $n_b \in \{1, 2, \cdots, N\}$, and $n_b \neq n_{b'}$, iff $b \neq b'$. We will discuss that the MSE of estimation depends on the bit-sensor assignment and propose an algorithm to find the optimum assignment that minimizes the MSE.

Suppose that N local binary quantization patterns, i.e., $\mathcal{P}_b$; $1 \leq b \leq N$, are obtained based on Gray labeling (see Chapter 5). For each b, $1 \leq b \leq N$, we assign $\mathcal{P}_b$ to sensor n_b. At sensor n_b, depending on its measurement Y_{n_b}, $\mathcal{P}_b$ decides the value of the bth bit in the N-bit quantization. If there are no measurement noises each sensor will quantize X to one of the bits of the N-bit word assigned to the division of X. Consequently, the division of X can be recognized correctly at the FC, resulting in the estimation MSE of $\Delta^2/12$. However, since each measurement is a noisy version of X, any of the quantized bits can be wrong, causing higher estimation error.

In inhomogeneous WSNs, the final estimation MSE depends on the bit-sensor assignment. Each particular assignment, i.e., $\pi = \{n_1, n_2, \cdots, n_N\}$, is a permutation of the numbers 1 to N. The assignment π means that bit 1 is assigned to sensor n_1, bit 2 is assigned to sensor n_2, etc. We wish to find

$$\pi^* = \arg\min_{\pi} \psi(\pi), \tag{6.1}$$

where $\psi(\pi)$ is the estimation MSE as a function of the bit-sensor assignment. In the distributed quantization based on the Gray labeling, the MSE does not have a closed form in terms of π. Therefore, to find the optimum bit-sensor assignment π^* one needs to work out the MSE as a function of $\sigma_1, \cdots, \sigma_N$ for every possible π, and then find the minimum of all MSEs. This approach is very cumbersome and requires $N!$ complicated computations or simulations of the MSE.

If we can write the MSE as a summation of N terms in a way that the bth term,

$1 \leq b \leq N$, depends only on pair (b, n_b), we can apply the Hungarian method [61] to find the optimum assignment with complexity order of $O(N^3)$. The Hungarian method can be used for the following problem. Assume that there are N tasks and N workers, and $N \times N$ pair costs $\Gamma_{b,n}$, $1 \leq b \leq N$, $1 \leq n \leq N$, showing the cost of assigning task b to worker n. The sum of costs for assignment $\pi = \{n_1, n_2, \cdots, n_N\}$ is defined as

$$\mathcal{C}(\pi) = \sum_{b=1}^{N} \Gamma_{b,n_b}. \tag{6.2}$$

The Hungarian method finds π^* that minimizes (6.2). In the next section, we approximate the estimation MSE to formulate it as (6.2).

6.3 Formulating the MSE

Since X is a random variable, the MSE is defined as

$$\begin{aligned} \psi &= E\{(X - \hat{X})^2\} = E\{E\{(X - \hat{X})^2 | X\}\} \\ &= \int_X \int_{\hat{X}} (X - \hat{X})^2 p(\hat{X}, X) dX d\hat{X}. \end{aligned} \tag{6.3}$$

If all N bits are correctly generated, $\hat{X}$ will be in the same division as X. However, if one or more bits are wrong, $\hat{X}$ could be in a different division. To formulate the MSE we should consider all cases where any combination of the N bits are wrong, and compute $\hat{X}$ and the probability function $p(\hat{X}, X)$ for that case. We derive an approximation for the MSE, by considering a high probable subset of these cases. In particular, we consider the cases that at most one of the N bits is wrong. This is a reasonable approximation when bit error probabilities are independent conditioned on X, which is followed from the fact that sensors' measurement noises are independent. In section 6.5, we investigate this approximation for different average SNRs.

Let β be the event where all N bits are correct, and γ_b, $1 \leq b \leq N$, be the event where only bit b is wrong. We can write an approximation for MSE as

$$\begin{aligned} \psi \simeq & \int_X \int_{\hat{X}} (X - \hat{X})^2 p(\hat{X}, X, \beta) dX d\hat{X} \\ & + \sum_b \int_X \int_{\hat{X}} (X - \hat{X})^2 p(\hat{X}, X, \gamma_b) dX d\hat{X}. \end{aligned} \tag{6.4}$$

For a uniform quantization and a uniformly distributed X the first term is equal to $\Delta^2/12 = 1/3(V/2^N)^2$. In the event of γ_b, knowing the labeling method, $\hat{X}$ can be

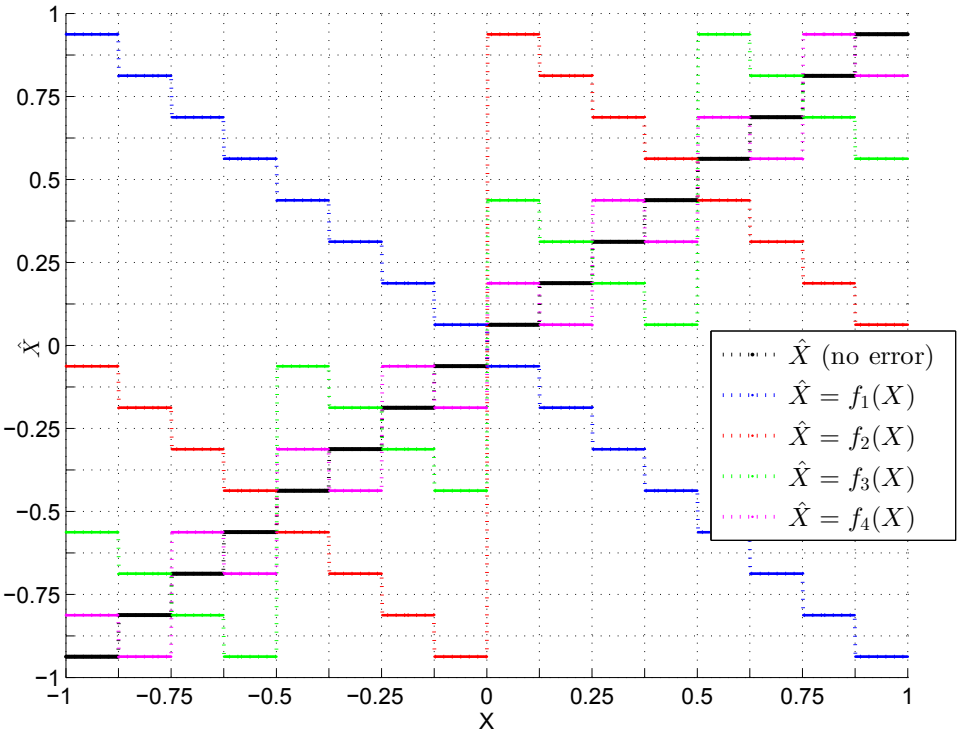

Figure 6.1: Functions $f_b(X)$ for $N=4$, using Gray labeling.

written as a function of X which depends on b, i.e., $\hat{X} = f_b(X)$. Fig. 6.1 shows these functions for $N = 4$, when the Gray labeling scheme is used. Substituting $f_b(X)$ in (6.4), we have

$$\psi \simeq \frac{1}{3}(\frac{V}{2^N})^2 + \frac{1}{2V}\int_X \sum_b (X - f_b(X))^2 p(\gamma_b|X) dX, \tag{6.5}$$

where $p(\gamma_b|X)$ is the probability that only bit b is wrong conditioned on X. Now, we approximate $p(\gamma_b|X)$ with the probability that bit b is wrong conditioned on X, i.e., $p(\alpha_b|X)$. This is a reasonable approximation when the probability that only bit b is wrong is bigger than the probability that bit b and some other bits are wrong. Hence, the approximation is close when sensors' measurement noises are not very high, and the network size in not very large. An example of a good condition is when $p(\alpha_b|X) \ll 1/N$.

The probability $p(\alpha_b|X)$ depends on $\mathcal{P}_b$ and the noise pdf of sensor n_b, i.e., $p(\alpha_b|X) = h_{b,n_b}(X)$. For Gaussian measurement noise this can be written as a sum of Q-functions that depend on $\mathcal{P}_b$ and σ_{n_b} (See Appendix C for details). Substituting $p(\gamma_b|X) \simeq p(\alpha_b|X)$ into (6.5) and computing the integral over X, MSE can be

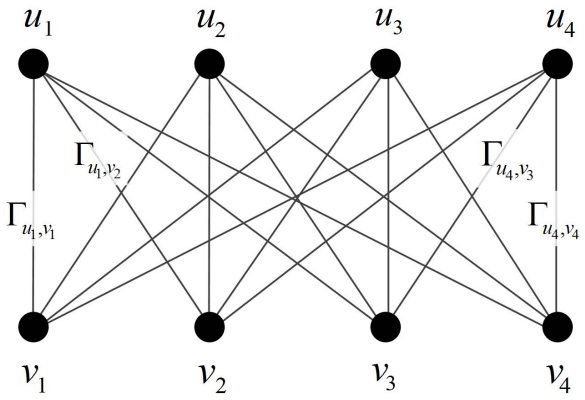

Figure 6.2: A complete bipartite graph.

summarized as

$$\psi(\pi) \simeq A + \sum_{b=1}^{N} \Gamma_{b,n_b}, \tag{6.6}$$

where $A = 1/3(V/2^N)^2$, and

$$\Gamma_{b,n_b} = \frac{1}{2V} \int_X (X - f_b(X))^2 h_{b,n_b}(X) dX. \tag{6.7}$$

Investigating (6.6) we notice that except for the constant term A, (6.6) is the sum of pair costs similar to (6.2), hence, to find the optimum assignment π^* we use the Hungarian method. The Hungarian algorithm can be implemented with complexity as low as $O(N^3)$ using Blossom algorithm [64] or Hopcroft-Karp algorithm [63].

6.4 Hungarian Algorithm for Bit-Sensor Assignment

Since we have N sensors and N bits, there are N^2 possible values of $\Gamma_{b,n}$. Given N^2 values of $\Gamma_{b,n}$, the Hungarian method is used to find π^* that minimizes (6.6). Assume a complete bipartite graph $G(U, V, W)$, where $U = \{u_1, u_2, \cdots, u_N\}$ is a set of vertices representing the N bits (the N quantizers), $V = \{v_1, v_2, \cdots, v_N\}$ is another set of vertices representing the N sensors, and W is the set of weights for each edge (u, v), connecting vertex $u \in U$ to vertex $v \in V$. The weight of edge (u, v) is $\Gamma_{u,v}$, see Fig. 6.2. To find the optimum bit-sensor assignment that results in the minimum MSE, we need to find a perfect matching of G which has the minimum sum of weights (Please see Section 2.3 for the details). This can be achieved using the Hungarian algorithm [61] shown in Algorithm 8. For details of the algorithm please see Section 2.3 and [61].

Algorithm 8 Hungarian method for best assignment

input: A bipartite graph $G(U, V, W)$, where U is the set of vertices representing the bits, V is the set of vertices representing the sensors, and W is the set of weights $\Gamma_{u,v}$, where $u \in U$ and $v \in V$.

Initialize a feasible vertex labeling l.
Determine the equivalent graph G_l.
Pick any matching M in G_l
while M is not perfect **do**
  if an augmenting path for M is found in G_l **then**
    Upgrade M to increase its size.[1]
  end if
  if no augmenting path exists **then**
    Improve l, and find the new G_l.
  end if
end while

6.5 Simulation Results

We first explain the performance of our algorithm for a sample case of $N = 8$. The measurement noise variances of sensors are shown in Table 6.1. A possible bit-sensor assignment is to assign the highest-precision bit to the sensor with smallest noise variance, the second high-precision bit to the sensor with the second smallest noise variance, and so on. This is shown in Table 6.1 as Assign I. A second possible bit-sensor assignment is the reverse form, i.e., to assign the highest-precision bit to the sensor with the highest noise variance, the second high-precision bit to the sensor with the second highest variance, and so on. This is shown as Assign II in the table. The optimum bit-sensor assignment based on our proposed algorithm is also shown in the table.

The estimation performance of different bit-sensor assignments are compared in terms of MSE. Both the simulation results and the approximate MSE calculation based on (6.6) are shown in the table. It can be seen that *i*) the approximate MSE formula of (6.6) used in our algorithm is very close to the simulated MSEs, *ii*) The performance of the optimum assignment resulted by our algorithm is better than both other assignment cases.

The performance gain of employing the optimum assignment compared to other assignments is more when there is higher inhomogeneity. The inhomogeneity can be described as the variance of σ_n^2, i.e., $\Lambda_{\sigma^2} = E\{(\sigma_n^2)^2 - \mu_{\sigma^2}^2\}$, where μ_{σ^2} is the mean of σ_n^2 defined as $E\{\sigma_n^2\} = 1/N \sum_n \sigma_n^2$. For $N = 4$, Fig. 6.3 shows the performance

Sensors' Measurement Noise Variances

n	1	2	3	4	5	6	7	8
σ_n	0.04	0.03	0.02	0.01	0.008	0.006	0.005	0.001

Different Bit-Sensor Assignments

n_b	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8
Assign I	1	2	3	4	5	6	7	8
Assign II	8	7	6	5	4	3	2	1
Optimum	5	6	7	8	4	3	2	1

Estimation MSE

	MSE (simulation)	MSE (approximate)
Assign I	0.0392	0.0374
Assign II	0.0098	0.0101
Optimum	0.0067	0.0068

Table 6.1: An example of bit-sensor assignment with $N = 8$.

of our algorithm, i.e., MSE_{alg}, versus $\sqrt{\Lambda_{\sigma^2}}$, and compares it with other assignment solutions. At each Λ_{σ^2}, the performance indicators have been averaged over 100 setups, where for each setup, σ_n^2, $1 \leq n \leq 4$, are randomly selected. However, for all 100 setups $\mu_{\sigma^2} = 0.03$, and Λ_{σ^2} is kept constant.

At each specific Λ_{σ^2}, all $N!$ assignments are tested and the average of their MSE is denoted as MSE_{avg}. As expected, by increasing Λ_{σ^2}, we see a larger gap between MSE_{avg} and MSE_{alg}. To evaluate the performance, we have included the minimum MSE of all assignments found through brute-force search, i.e., MSE_{min}. The small deference between MSE_{alg} and MSE_{min} is due to the approximation of Section 6.3. The proximity of MSE_{alg} and MSE_{min} provides another evidence for the accuracy of our algorithm based on MSE approximation. We have also shown the performance of Assign II solution as MSE_{II} in Fig. 6.3. It can be seen that for all inhomogeneities, the solution obtained through our algorithm performs better than Assign II. Fig 6.4 shows the same properties versus μ_{σ^2}, when $\Lambda_{\sigma^2} = 0.015^2$.

6.6 Conclusion

We have addressed the problem of bit-sensor assignment for binary distributed quantization in an inhomogeneous network of N sensors. Our algorithms is based on approximating the MSE in order to formulate it as a sum of N bit-sensor pair costs. This formulation allows us to use the Hungarian method for finding the best

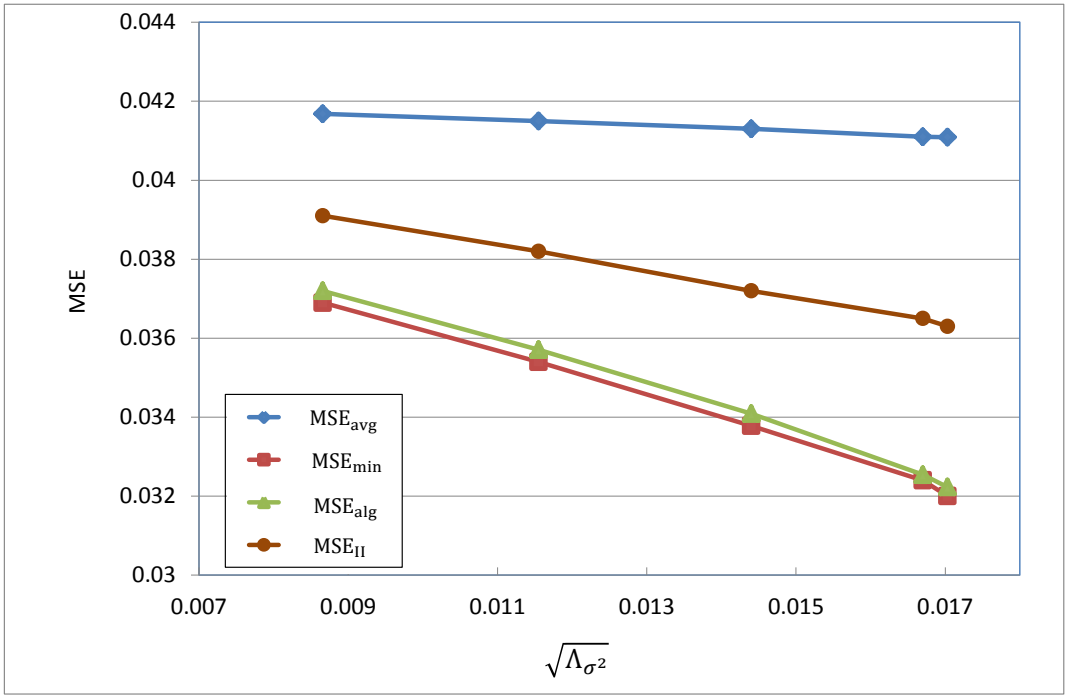

Figure 6.3: Performance of our algorithm versus $\sqrt{\Lambda_{\sigma^2}}$.

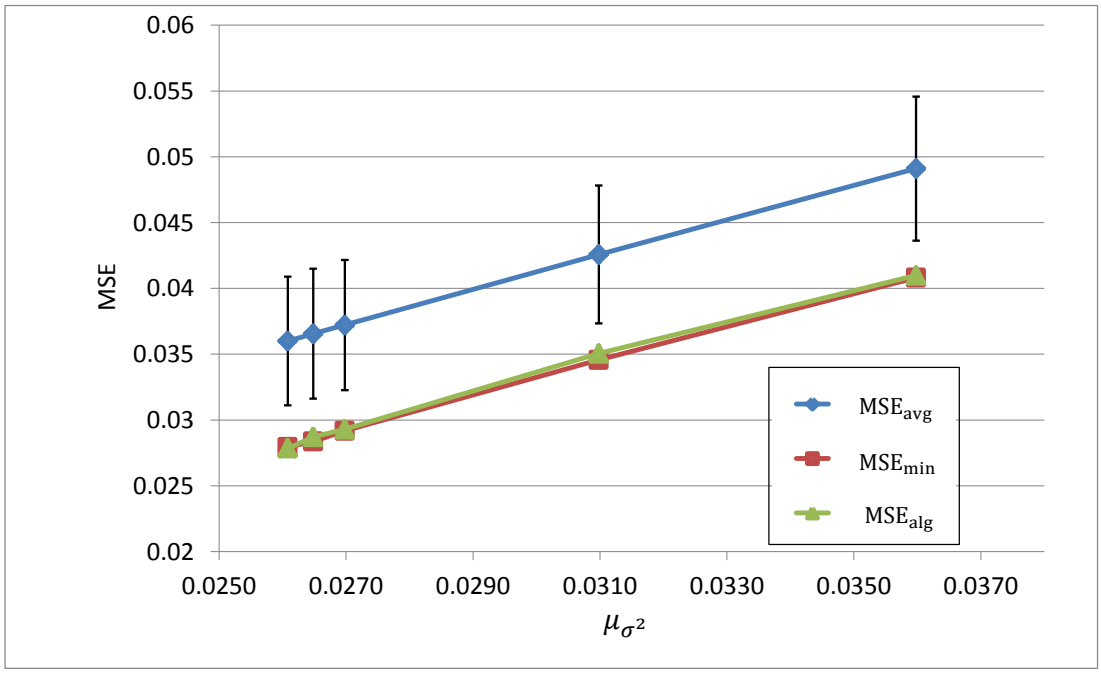

Figure 6.4: The MSE of brute-force and the Hungarian algorithm versus μ_{σ^2}.

assignment that minimizes the MSE with $O(N^3)$ complexity rather than $O(N!)$. The simulation results show significant gains when using the optimal assignment compared to simple-order or arbitrary assignments. The studies in this chapter are published in [106].

Chapter 7

Conclusion and Future Work

In this thesis, we addressed estimation and quantization problems in wireless sensor networks, and developed some distributed algorithms for these applications. Here, we summarize the contributions of each chapter. To conclude, we suggest some possible research trends that can be done in the future.

7.1 Conclusion

Chapter 3 was focused on stochastic inference in WSNs with no fusion centre. In this chapter, we formulated a non-parametric solution, by mixing the sum-product algorithm with the particle filtering technique, for the general non-Gaussian and nonlinear inference problems in WSNs.

We first used the factor graph to model the stochastic dependencies between the variables involved in the inference problem. To estimate the unknown variables of the problem a message passing algorithm, called the sum-product algorithm, is run on the factor graph. However, except for problems that only involve binary or discrete variables or Gaussian-distributed continuous variables, the message update rules specified by the sum-product algorithm could be very complex in general. Therefore, to maintain a feasible complexity for the general non-Gaussian, nonlinear problems, the sum-product update rules were reformulated using the particle filtering. In other words, using Monte Carlo method and importance sampling, we solved the analytical intractable integrals of the sum-product algorithm for continuous variables, with the complexity which is independent of the order of the integral. To achieve that, the messages, which are generally continuous pdfs, were represented by a set of random samples and their importance weights. At variable nodes and function nodes of the FG, we formulated the rules to update the samples and their

importance weights.

Our proposed particle-based message passing algorithm is applicable to many estimation problems in WSNs, e.g. localization, tracking, link monitoring, etc. For each of these applications, we need to first determine the appropriate factor graph. Then, the particle-based message passing algorithm decides each sensors' processing task. To implement our algorithm, we considered the problem of distributed target tracking and the problem of multi-target tracking with data association in WSNs. Because of considering the acceleration parameter in the target's dynamic model as well as the quantization of measurements the posterior pdfs and consequently the factor graph messages would be non-Gaussian. Therefore, the particle-based message passing algorithm was used and its performance was studied based on the number of quantization bits, the number of particles and the measurement noise.

We compared our particle-based message passing algorithm with two other algorithms. The extended Kalman filter and the Gaussian message passing algorithm. The results showed that the estimation MSE of our algorithm was better than the other two methods. Our particle-based message passing algorithm achieved reliable tracking results even when the number of quantized levels were as small as 4 and 8, with only 5 particles for representing the messages. The particle-based message passing algorithm is superior to the existing methods also in the sense that it assumes no limitations on the type of relations between the variables, neither it restricts to Gaussian noise assumptions. It is a low-cost algorithm in the sense of communication costs, which requires the sensors to broadcast only a low-bit quantized version of their data in their neighborhood, instead of the high-precision raw values.

In Chapter 4, we focused on parameter estimation in WSNs with constrained communication capabilities. In such scenarios, sensors' noisy measurements must be quantized locally before transmitting to the fusion centre, where the unknown parameter is estimated. In this chapter, we developed an algorithm to jointly design N local quantizers given the pdf of the unknown parameter.

We designed the local quantizers by maximizing the mutual information between the unknown parameter and the quantizers' outputs. We first formulated the mutual information based on the N quantization functions. To find the nth quantization function, assuming the other quantizers are fixed, we used the double maxima approach to iteratively maximize the mutual information. In Section 4.5,

we incorporated the effect of the communication channels in the design of optimal quantizers, and designed channel-aware quantization rules.

Our design method has complexity benefits over the other methods. Unlike other optimization measures, the mutual information measure allows the global optimization criterion to be broken down into smaller criteria. In fact, by basing our algorithm on the relaxed problem in Section 4.3 we reduced the dimension of integrals, hence decreased the complexity. This was consistent with the simulation results. Furthermore, our algorithm allows the design complexity to be significantly reduced when the sensors' measurements are conditionally independent, which is a common assumption in sensor networks.

Using our algorithm one could design the quantizers independent of the choice of the decoder or estimator in the fusion centre. This enables the designer to obtain a quantization algorithm useful for a range of applications such as estimation, detection, classification or feature extraction. We have shown this through simulating an estimation and a detection application, where our method achieved comparable estimation and detection errors to the methods that have been specifically designed for those applications. The channel-aware quantizers were designed for the binary symmetric channel. We observed that the optimal local quantizers in the presence of non-ideal channels are different from the local quantizers that are optimized without considering the channel effect.

In Chapter 5, we focused on parameter estimation in WSNs with very stringent energy limitations, where only one bit quantization per observation is tolerated. Unlike the iterative algorithm of Chapter 4, which achieves a local optimum solution, in this chapter we aimed to find a global solution to the quantization problem. Since sensors are only allowed one bit per measurement finding the global optimum solution to the design problem largely affects the estimation performance.

We viewed the problem as if a set of B local binary quantizers together imitate a B-bit quantization. Then, we used an algebraic approach to describe the distributed quantization. Since an $L = 2^B$ level quantizer can assign a B-bit binary number to each partition, a set of B binary quantizers can be obtained by mapping each of the B bits to a local quantizer. In Chapter 5, we showed that besides the default mapping, other sets of B binary quantizers can be obtained by searching all $B \times B$ binary matrices of rank B. The best of these choices that minimizes a performance metric, i.e., Cramér-Rao lower bound, was chosen as the solution. When $N >$

B measurements are available it is possible to estimate the unknown parameter with more accuracy. The redundant measurements enable some error correction possibility in estimating the unknown parameter. In this case, the optimum N binary quantizers where found by searching all $B \times N$ binary matrices of rank B. For scenarios where the search space size is too big, sub-optimal strategies were suggested in Section 5.4.4 to reduce the complexity.

To estimate the unknown parameter in the fusion centre, two decoders, i.e., the discrete and the continuous maximum likelihood decoders, were derived based on the discrete and continuous a posteriori distributions, respectively. Also, some analytical performance bounds based on the rate-distortion theory and Cramér-Rao lower bound where derived in Section 5.5.2. The performance of the method was studied through different simulations. The estimation MSE was compared with the performance bounds and other distributed quantization methods. We showed that our method outperforms other binary quantization methods, specifically for higher signal-to-noise ratios.

In Chapter 6, we addressed the binary quantization in inhomogeneous WSNs, where sensors do not have the same measurement quality. We showed that in such conditions, the order of assigning the local binary quantizers to the sensors affects the estimation performance. We proposed an algorithm based on Hungarian method to find the best assignment that minimizes the MSE of estimation. We first approximated and formulated the MSE as a summation of N terms, each being a function of only one sensor's measurement noise. Consequently, we were able to use the Hungarian method to find the optimum bit-sensor assignment. Through simulations we confirmed that the assignment found by our algorithm has the MSE performance which is very close to the optimum performance, and is superior to heuristic assignment solutions.

7.2 Future Work

Particle-based Message Passing Algorithm for Other Applications

In our particle-based message passing algorithm introduced in Chapter 3, we combined two important tools, i.e. the sum-product algorithm and the particle filtering, to enhance the functionality of both. By using particle-based message passing algorithm, we extended the ability of the FG to efficiently cover many non-Gaussian,

nonlinear frameworks with feasible complexity. Therefore, the particle-based message passing algorithm is a powerful framework that can be used in wide range of stochastic inference problems, not limited to WSNs. The advantage of this method can be noticed more in problems with complicated stochastic relations between variables, including those involving nonlinear relations and non-Gaussian noises. The factor graph helps model these relations and the particle-based message passing algorithm solves the analytically untraceable message update rules to enable stochastic inference. These problems can appear in various disciplines of science, including geostatistics, finance, and economics. For example, in geostatistics, the particle-based message passing algorithm can be used in seismic interpretation and reservoir modeling. In these applications, the relations in the state-space model and the relations between the measurements and the states can be very complex, therefore, the particle-based message passing algorithm is advantageous. To address such problems, first, an appropriate stochastic model must be derived. The choice of the model variables and a proper stochastic factorization affects the structure of the resulting factor graph. Subsequently, the message processing rules can be derived based on the particle-based message passing algorithm to find the marginal beliefs of the state variables.

Extending the Design Method based on Mutual Information

In Chapter 4, an algorithm based on mutual information was suggested for designing optimum distributed quantizers. The method was only practiced for scalar parameters. In many situations, a vector of parameters needs to be quantized at each sensor rather than a scalar parameter. The rate-distortion theory states that the joint quantization of the variables is more efficient than separately quantizing the single variables. Therefore, deriving optimum distributed quantizers can be extended to the problem of vector quantization, and a framework based on mutual information can be derived for that.

Also, in Chapter 4, we derived channel-aware distributed quantizers by assuming binary symmetric channel model. This is a basic assumption for a communication channel. Taking similar principles, a method can be devised for more advanced channel models. As long as the channels are independent and each quantization-plus-channel branch can be modeled as a first-order Morkov chain, the steps are similar to those introduced in Chapter 4.

Combining the Global and Local Methods

In Chapter 5, we found the global solution for the set of local binary quantizers. However, in that Chapter, we had limited the quantization edges to be chosen form a predetermined list of $L-1$ fixed edges. Therefore, although the solution is globally optimum among all sets of quantizers that can be represented as a $B \times B$ binary matrix, the position of the quantization edges can still be improved. Now, we can use the method of Chapter 4 to fine-tune the quantizers. In a few words, starting with the set of quantizers resulted from the method of Chapter 5, the iterative method based on maximizing the mutual information can find the local optimums.

Bibliography

[1] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. Wiley-Interscience, 2007.

[2] P. Baronti, P. Pillai, V. Chook, S. Chessa, A. Gotta, and Y. Hu, "Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards," *Computer Communications*, vol. 30, no. 7, pp. 1655–1695, May 2007.

[3] R. Gershenfeld, N. Krikorian and D. Cohen, "The internet of things," *Scientific American*, vol. 291, no. 4, pp. 76–81, Oct. 2004.

[4] D. Evans, "The internet of things: How the next evolution of the internet is changing everything," *CISCO white paper*, p. 111, Apr. 2011.

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

[6] E. F. Nakamura, A. A. F. Loureiro, and A. C. Frery, "Information fusion for wireless sensor networks: Methods, models, and classifications," *ACM Computing Surveys*, vol. 39, no. 3, Sept. 2007.

[7] M. Ahmed, "Decentralized information processing in wireless peer to peer networks," Ph.D. dissertation, UCLA Electrical Engineering Department., 2002.

[8] A. Ribeiro, G. Giannakis, and S. Roumeliotis, "SOI-KF: distributed Kalman filtering with low-cost communications using the sign of innovations," *IEEE Transactions on Signal Processing*, vol. 54, no. 12, pp. 4782–4795, Dec. 2006.

[9] A. Oka and L. Lampe, "Distributed scalable multi-target tracking with a wireless sensor network," in *IEEE International Conference onCommunications*, 2009, pp. 1–6.

[10] J. Liu, M. Chu, and J. Reich, "Multitarget tracking in distributed sensor networks," *Signal Processing Magazine, IEEE*, vol. 24, no. 3, pp. 36 –46, may 2007.

[11] Y. Mao, F. Kschischang, B. Li, and S. Pasupathy, "A factor graph approach to link loss monitoring in wireless sensor networks," *IEEE Journal on Selected Areas in Communication*, vol. 23, pp. 820–829, 2005.

[12] W. Wei, C. Wan-Dong, W. Bei-Zhan, W. Ya-Ping, and T. Guang-Li, "The factor graph approach for inferring link loss in MANET," in *International Conference on Internet Computing in Science and Engineering*, Jan. 2008, pp. 166–172.

[13] J.-J. Xiao, A. Ribeiro, Z.-Q. Luo, and G. Giannakis, "Distributed compression-estimation using wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 27 – 41, july 2006.

[14] J. Fang and H. Li, "Distributed adaptive quantization for wireless sensor networks: From delta modulation to maximum likelihood," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 5246 –5257, Oct. 2008.

[15] J. Li and G. AlRegib, "Distributed estimation in energy-constrained wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 57, no. 10, pp. 3746–3758, oct 2009.

[16] A. Ribeiro and G. Giannakis, "Bandwidth-constrained distributed estimation for wireless sensor networks-part I: Gaussian case," *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 1131–1143, Mar. 2006.

[17] ——, "Bandwidth-constrained distributed estimation for wireless sensor networks-part II: unknown probability density function," *IEEE Transactions on Signal Processing*, vol. 54, no. 7, pp. 2784 –2796, 2006.

[18] B. Chen, L. Tong, and P. Varshney, "Channel-aware distributed detection in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 16–26, july 2006.

[19] A. Willig, "Wireless sensor networks: concept, challenges and approaches," *e&i Elektrotechnik und Informationstechnik*, vol. 123, no. 6, pp. 224–231, 2006.

[20] N. Srivastava, "Challenges of next-generation wireless sensor networks and its impact on society," *Journal of Telecommunications*, vol. 1, pp. 128–133, Feb 2010.

[21] M. Gastpar, M. Vetterli, and P. Dragotti, "Sensing reality and communicating bits: a dangerous liaison," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 70–83, July 2006.

[22] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

[23] J. Max, "Quantizing for minimum distortion," *IRE Transactions on Information Theory*, vol. 6, no. 1, pp. 7–12, Mar. 1960.

[24] T. Fine, "Optimum mean-square quantization of a noisy input (corresp.)," *IEEE Transactions on Information Theory*, vol. 11, no. 2, pp. 293–294, Apr. 1965.

[25] E. Ayanoglu, "On optimal quantization of noisy sources," *IEEE Transactions on Information Theory*, vol. 36, no. 6, pp. 1450–1452, Nov. 1990.

[26] W. Lam and A. Reibman, "Quantizer design for decentralized estimation systems with communications constraints," in *In Proceedings of the 23rd Annual Conference on Information Sciences and Systems*, Mar. 1989, pp. 489–494.

[27] J. Gubner, "Distributed estimation and quantization," *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 1456 –1459, July 1993.

[28] W. Lam and A. Reibman, "Design of quantizers for decentralized estimation systems," *IEEE Transactions on Information Theory*, vol. 41, no. 11, pp. 1602–1605, Nov. 1993.

[29] M. Longo, T. Lookabaugh, and R. Gray, "Quantization for decentralized hypothesis testing under communication constraints," *IEEE Transactions on Information Theory*, vol. 36, no. 2, pp. 241–255, Mar. 1990.

[30] I. Csiszar and T. G., "Information geometry and alternating minimization procedures," *Statistics and Decisions, Supplement Issue*, 1984.

[31] Z.-Q. Luo, "An isotropic universal decentralized estimation scheme for a bandwidth constrained ad hoc sensor network," *IEEE J. Select. Areas Commun.*, vol. 23, no. 4, pp. 735–744, 2005.

[32] E. Msechu, S. Roumeliotis, A. Ribeiro, and G. Giannakis, "Decentralized quantized kalman filtering with scalable communication cost," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3727–3741, 2008.

[33] R. Niu and P. Varshney, "Target location estimation in sensor networks with quantized data," *IEEE Transactions on Signal Processing*, vol. 54, no. 12, pp. 4519–4528, 2006.

[34] Z.-Q. Luo, "Universal decentralized estimation in a bandwidth constrained sensor network," *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 2210–2219, 2005.

[35] Z.-Q. Luo and J.-J. Xiao, "Universal decentralized estimation in a bandwidth constrained sensor network," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, vol. 4, 2005, pp. 829–832.

[36] J.-J. Xiao and Z.-Q. Luo, "Decentralized estimation in an inhomogeneous sensing environment," *IEEE Transactions on Information Theory*, vol. 51, no. 10, pp. 3564–3575, 2005.

[37] H. Papadopoulos, G. Wornell, and A. Oppenheim, "Sequential signal encoding from noisy measurements using quantizers with dynamic bias control," *IEEE Transactions on Information Theory*, vol. 47, no. 3, pp. 978–1002, Mar. 2001.

[38] J. Fang and H. Li, "Optimal/near-optimal dimensionality reduction for distributed estimation in homogeneous and certain inhomogeneous scenarios," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4339–4353, 2010.

[39] *Handbook of Statistics, Volume 25: Bayesian Thinking Modeling and Computation*, 2005.

[40] H. Mitchell, *Multi-sensor data fusion: an introduction.* Springer, 2007.

[41] P. Fearnhead, "Sequential monte carlo methods in filter theory," Ph.D. dissertation, University of Oxford, 1998.

[42] R. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME, Journal of Basic Engineering*, no. 82, pp. 35–45, 1960.

[43] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications.* Artech House, 2004.

[44] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[45] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, July 2000.

[46] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations, Tech. Rep. TR-2001-22, Jan. 2002.

[47] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[48] J. Dauwels, S. Korl, and H.-A. Loeliger, "Particle methods as message passing," in *Information Theory, 2006 IEEE International Symposium on*, July 2006, pp. 2052–2056.

[49] S. Movaghati and M. Ardakani, "Particle-based message passing algorithm for inference problems in wireless sensor networks," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 745–754, July 2010.

[50] ——, "Stochastic inference in wireless sensor networks," in *Building Sesnor Networks*, I. Nikolaidis, Ed. Florida, USA: CRC Press, 2013.

[51] E. Sudderth, A. Ihler, W. Freeman, and A. Willsky, "Nonparametric belief propagation," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2003, pp. 605–612.

[52] B. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman and Hall/CRC, 1986.

[53] T. M. Cover and J. A. Thomas, *Elements of information theory*. New York, NY, USA: Wiley-Interscience, 1991.

[54] C. E. Shannon, "A mathematical theory of communication," *Bell system technical journal*, vol. 27, 1948.

[55] I. Schizas, G. Giannakis, and N. Jindal, "Distortion-rate bounds for distributed estimation using wireless sensor networks," *EURASIP Journal on Advances in Signal Processing*, 2008.

[56] R. Gray, "Vector quantization," *IEEE ASSP Magazine*, vol. 1, no. 2, pp. 4–29, Apr. 1984.

[57] Y. Linde, A. Buzo, and R. Gray, *IEEE Transactions on Communications*.

[58] A. Papoulis and S. Pillai. NY, USA: McGraw-Hill, 2001.

[59] H. L. V. Trees, *Detection, Estimation, and Modulation Theory*. John Wiley and Sons, Inc., 2001.

[60] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2009.

[61] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, no. 2, p. 8397, 1955.

[62] J. Edmonds and R. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *Journal of the Association for Computing Machinery*, vol. 19, no. 2, pp. 248–264, Apr. 1972.

[63] J. E. Hopcroft and R. M. Karp, "An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs." *SIAM J. Comput.*, vol. 2, no. 4, pp. 225–231, 1973.

[64] J. Edmonds, "Paths, Trees and Flowers," *Canadian Journal of Mathematics*, vol. 17, pp. 449–467, 1965.

[65] N. Blum, "A simplified realization of the hopcroft-karp approach to maximum matching in nonbipartite graphs," Tech. Rep., 1999.

[66] B. Etzlinger, W. Haselmayr, and A. Springer, "Message passing methods for factor graph based mimo detection," in *Wireless Advanced (WiAd), 2011*, June 2011, pp. 132–137.

[67] H. Wymeersch, U. Ferner, and M. Win, "Cooperative Bayesian self-tracking for wireless networks," *IEEE Communication Letters*, vol. 12, no. 7, pp. 505–507, 2008.

[68] J. Lien, "A framework for cooperative localization in ultra-wideband wireless networks," Master's thesis, Massachusetts Institute of Technology, Cambridge, 2007.

[69] L. Mihaylova, D. Angelova, S. Honary, D. Bull, C. Canagarajah, and B. Ristic, "Mobility tracking in cellular networks using particle filtering," *IEEE Transactions on Wireless Communication*, vol. 6, no. 10, pp. 3589–3599, Oct. 2007.

[70] C. Kreucher, K. Kastella, and A. Hero, "Tracking multiple targets using a particle filter representation of the joint multitarget probability density," in *International Symposium on Optical Science and Technology*, 2003, pp. 258–259.

[71] A. Ihler, J. Fisher III, R. Moses, and A. Willsky, "Nonparametric belief propagation for self-calibration in sensor networks," in *Proceding of the 3rd international symposium on Information processing in sensor networks*, pp. 225–233, year =.

[72] A. Ihler, I. Fisher, J.W., R. Moses, and A. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 809–819, Apr. 2005.

[73] J. Williams, I. Fisher, J.W., and A. Willsky, "Optimization approaches to dynamic routing of measurements and models in a sensor network object tracking problem," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Mar. 2005, pp. 1061–1064.

[74] C. Wei, L. Xin, and C. Mei, "Cooperative distributed target tracking algorithm in mobile wireless sensor networks," in *5th International Conference on Computer Science and Education (ICCSE)*, aug 2010, pp. 120 –123.

[75] N. Sandell and R. Olfati-Saber, "Distributed data association for multi-target tracking in sensor networks," in *47th IEEE Conference on Decision and Control*, dec 2008, pp. 1085–1090.

[76] R. Gray and D. Neuhoff, "Quantization," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2325–2383, Oct. 1998.

[77] Y. Ephraim and R. Gray, "A unified approach for encoding clean and noisy sources by means of waveform and autoregressive model vector quantization," *IEEE Transactions on Information Theory*, vol. 34, no. 4, pp. 826–834, July 1988.

[78] A. Ribeiro and G. Giannakis, "Bandwidth-constrained distributed estimation for wireless sensor networks-part I: Gaussian case," *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 1131–1143, Mar. 2006.

[79] T. Aysal and K. Barner, "Constrained decentralized estimation over noisy channels for sensor networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1398–1410, Apr. 2008.

[80] Y. Lin, "Quantization for distributed detection under link outages," in *42nd Asilomar Conference on Signals, Systems and Computers*, 2008, pp. 1948–1952.

[81] H. Poor and J. Thomas, "Applications of ali-silvey distance measures in the design generalized quantizers for binary decision systems," *IEEE Transactions on Communication*, vol. 25, no. 9, pp. 893–900, Sept. 1977.

[82] H. Poor, "Fine quantization in signal detection and estimation," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 960–972, 1988.

[83] P. Venkitasubramaniam, L. Tong, and A. Swami, "Quantization for maximin ARE in distributed estimation," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3596–3605, July 2007.

[84] M. Fowler, M. Chen, and S. Binghamton, "Fisher-information-based data compression for estimation using two sensors," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 3, pp. 1131–1137, 2005.

[85] D. Messerschmitt, "Quantizing for maximum output entropy (corresp.)," *IEEE Transactions on Information Theory*, vol. 17, no. 5, p. 612, Sept. 1971.

[86] R. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE Transactions on Information Theory*, vol. 18, no. 4, pp. 460–473, July 1972.

[87] N. Farvardin and V. Vaishampayan, "Optimal quantizer design for noisy channels: An approach to combined source - channel coding," *IEEE Transactions on Information Theory*, vol. 33, no. 6, pp. 827–838, Nov. 1987.

[88] G. Fubine, "Sugli integrali multipli," *Opere scelte*, vol. 2, pp. 243–249, 1958.

[89] A. Kurtenbach and P. A. Wintz, "Quantizing for noisy channels," *IEEE Transactions on Communication Technology*, vol. 17, no. 2, pp. 291–302, Apr. 1969.

[90] N. Farvardin and V. Vaishampayan, "On the performance and complexity of channel-optimized vector quantizers," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 155–160, 1991.

[91] T. Duman and M. Salehi, "Decentralized detection over multiple-access channels," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 2, pp. 469–476, Apr. 1998.

[92] B. Liu and B. Chen, "Channel-optimized quantizers for decentralized detection in sensor networks," *IEEE Transactions on Information Theory*, vol. 52, no. 7, pp. 3349–3358, July 2006.

[93] N. Wernersson, J. Karlsson, and M. Skoglund, "Distributed quantization over noisy channels," *IEEE Transactions on Communication*, vol. 57, no. 6, pp. 1693–1700, June 2009.

[94] M. Valipour and F. Lahouti, "Channel optimized distributed multiple description coding," *IEEE Transactions on Signal Processing*, vol. 60, no. 5, pp. 2539–2551, May 2012.

[95] S. M. Ali and S. D. Silvey, "A general class of coefficients of divergence of one distribution from another," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 28, no. 1, pp. 131–142, 1966.

[96] "Distributed channel-aware quantization based on maximum mutual information," *submitted to IEEE Transactions on Information Theory*, Feb. 2014.

[97] J. Xiao, S. Cui, Z. Luo, and A. Goldsmith, "Power scheduling of universal decentralized estimation in sensor networks," *IEEE Transactions on Signal Processing*, vol. 54, no. 2, pp. 413–422, Feb. 2006.

[98] A. Krasnopeev, J. Xiao, and Z. Luo, "Minimum energy decentralized estimation in a wireless sensor network with correlated sensor noises," *EURASIP Journal on Wireless Communication and Networking*, no. 4, pp. 473–482, Sept. 2005.

[99] F. Chapeau-Blondeau and D. Rousseau, "Noise-enhanced performance for an optimal Bayesian estimator," *IEEE Transactions on Signal Processing*, vol. 52, no. 5, pp. 1327–1334, may 2004.

[100] G. Balkan and S. Gezici, "Crlb based optimal noise enhanced parameter estimation using quantized observations," *IEEE Signal Processing Letter*, vol. 17, no. 5, pp. 477–480, May 2010.

[101] H. Li and J. Fang, "Distributed adaptive quantization and estimation for wireless sensor networks," *IEEE Signal Processing Letter*, vol. 14, no. 10, pp. 669–672, Oct. 2007.

[102] R. Gray, S. Boyd, and T. Lookabaugh, "Low rate distributed quantization of noisy observations," in *Allerton Conference on Communication, Control, and Computing*, 1985, pp. 354–358.

[103] S. Lin and D. J. Costello, *Error Control Coding, Second Edition.* Prentice-Hall, Inc., 2004.

[104] S. Movaghati and M. Ardakani, "Energy-efficient quantization for parameter estimation in inhomogeneous wireless sensor networks," in *IEEE Global Communication Conference GLOBCOM*, Houston, Texas, USA, Dec. 2011, pp. 1–5.

[105] ——, "Distributed binary quantization of a noisy source in wireless sensor networks," *submitted to International Journal of Distributed Sensor Networks*, Sept. 2013.

[106] ——, "Optimum bit-sensor assignment for distributed estimation in inhomogeneous sensor networks," *IEEE Communication Letters*, no. 99, pp. 1–4, Feb. 2014.

[107] C. Fox, *An Introduction to the Calculus of Variations.* NY, USA: Dover, 1987.

[108] T. Berger, *Rate Distortion Theory: Mathematical Basis for Data Compression.* Englewood Cliffs, NJ, USA: Prentice Hall, 1971.

[109] E. Masazade, R. Niu, P. Varshney, and M. Keskinoz, "Energy aware iterative source localization for wireless sensor networks." *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4824–4835, 2010.

Appendix A

Proofs for Chapter 4

A.1 Double Maxima Method

Proof of i and ii. To prove i, it suffices to show that, under the constraint $\int_X f(X, Z_{1:n}) = 1$,

$$\max_f \mathcal{I}(p, f) = \mathcal{I}(X; Z_{1:n}). \tag{A.1}$$

To evaluate that, using the method of Lagrange multipliers, we need to solve

$$\frac{\partial}{\partial f(X, Z_{1:n})} \Big\{ \mathcal{I}(p, f(X, Z_{1:n})) + \sum_{Z_{1:n}} \lambda(Z_{1:n}) \Big[\int_X f(X, Z_{1:n}) - 1 \Big] \Big\} = 0. \tag{A.2}$$

Substituting $\mathcal{I}(p, f)$ from (4.12) and taking the above derivative, we get[1].

$$\int_{Y_{1:n}} p(X, Y_{1:n}) p(Z_{1:n-1}|Y_{1:n-1}) p(Z_n|Y_n) \frac{1}{f(X, Z_{1:n})} + \lambda(Z_{1:n}) = 0. \tag{A.3a}$$

$$\int_X f(X, Z_{1:n}) = 1 \tag{A.3b}$$

After finding $\lambda(Z_{1:n})$ according to the above two equations, we have

$$f^*(X, Z_{1:n}) = \frac{\int_{Y_{1:n}} p(X, Y_{1:n}) p(Z_{1:n}|Y_{1:n})}{\int_X \int_{Y_{1:n}} p(X, Y_{1:n}) p(Z_{1:n}|Y_{1:n})} = p(X|Z_{1:n}). \tag{A.4}$$

Furthermore, the second derivative of $\mathcal{I}(p, f)$ with respect to f is negative, hence $\mathcal{I}(p, f)$ reaches its maximum at f^*, which proves *ii*. Substituting f^* in $\mathcal{I}(p, f)$, we can easily verify that $\mathcal{I}(p, f^*) = \mathcal{I}(X; Z_{1:n})$ and hence (A.1) is proved. □

Proof of iii. Finding the optimal p^* that maximizes $\mathcal{I}(p, f)$ is equivalent to finding the optimal function $p^*(Z_n|Y_n = y)$ that maximizes $\mathcal{I}(p(Z_n|Y_n = y), f)$, for every

[1]For details of how (A.3) is derived from (A.2), please refer to Euler-Lagrange equations in variational calculus [107].

$y \in \mathbb{R}^2$. Substituting $p(Z_n|Y_n = y)$ from (4.10), this means for every $y \in \mathbb{R}$ we must find,

$$\begin{aligned}
&\max_{p(Z_n|Y_n=y)} \mathcal{I}(p(Z_n|Y_n = y), f) \\
&\qquad = \max_{l} \int_X \int_{Y_{1:n-1}} \sum_{Z_{1:n}} p(X, Y_{1:n-1}, y) p(Z_{1:n-1}|Y_{1:n-1}) \times \delta_{lZ_n} \times \log f(X, Z_{1:n}) \\
&\qquad = \max_{l} \int_X \int_{Y_{1:n-1}} \sum_{Z_{1:n-1}} p(X, Y_{1:n-1}, y) p(Z_{1:n-1}|Y_{1:n-1}) \times \log f(X, Z_{1:n-1}, l),
\end{aligned} \tag{A.5}$$

where the last equality is the result of the property $\sum_a \delta_{a_0 a} f(a) = f(a_0)$. Hence, *iii* is resulted. □

[2] According to (4.10), $p(Z_n|Y_n = y_1) = \delta_{l_1 Z_n}$ and $p(Z_n|Y_n = y_2) = \delta_{l_2 Z_n}$, where $l_1 = \mathcal{Q}_n(y_1)$ and $l_2 = \mathcal{Q}_n(y_2)$. Since the values of $\mathcal{Q}_n(y)$ at different points of Y_n do not depend on each other, $p(Z_n|Y_n = y_1)$ and $p(Z_n|Y_n = y_2)$ are also independent for every $y_1, y_n \in \mathbb{R}$; $y_1 \neq y_2$

Appendix B

Proofs for Chapter 5

B.1 Rate-Distortion Bound for Uniform Parameter

Here, we suggest an approximate lower bound for the MSE of distributed estimation problem from quantized measurements. The proposed lower bound can be numerically calculated for any distribution of the unknown scalar parameter. To derive the lower bound, we consider a centralized scalar quantization/estimation system with a rate distortion constraint of R. In this system, the noisy measurement $Y = X + w$ is first used to estimate X as $\hat{X}$ and then this estimation is quantized as $Q(\hat{X})$. The distortion (MSE) resulting from this centralized system, i.e., $D = E\{(X - Q(\hat{X}))^2\}$, can be used as a lower bound for our decentralized system. The MSE, can be approximated as

$$D \simeq \underbrace{E\{(X - \hat{X})^2\}}_{D_1} + \underbrace{E\{(\hat{X} - Q(\hat{X}))^2\}}_{D_2}. \tag{B.1}$$

In the above approximation, the cross term $E\{(X - \hat{X})(\hat{X} - Q(\hat{X}))\}$ is ignored from the right hand side, assuming an ideal quantizer, where quantization error is uncorrelated from the quantizer input. In the following, each of the two terms D_1 and D_2 are calculated.

The first term D_1 is by definition,

$$D_1 = \int_x \int_{\hat{x}} (x - \hat{x})^2 p_{X,\hat{X}}(x, \hat{x}) d\hat{x} dx. \tag{B.2}$$

To calculate the above integral, we need to find the joint pdf $p_{X,\hat{X}}(x, \hat{x})$. Notice that $\hat{x}$ is a deterministic function of the measurement value y, i.e., $\hat{x} = f(y)$. For example, if the MMSE estimator is used, $f(y) = E\{X|y\}$. Therefore, we have $p_{X,\hat{X}}(x, \hat{x}) = p_{\hat{X}|X}(f(y)|x) p_X(x)$. The term $p_{\hat{X}|X}(f(y)|x)$ can be calculated as

$$p_{\hat{X}|X}(f(y)|x) = \frac{p_{Y|X}(f^{-1}(\hat{x})|x)}{|f'(f^{-1}(\hat{x}))|}. \tag{B.3}$$

The conditional pdf $p_{Y|X}(y|x)$, or equivalently the measurement noise pdf, is assumed to be known. For example, for zero-mean Gaussian noise with variance σ^2, $p_{Y|X}(y|x) = \mathcal{N}(y; x, \sigma^2)$, where the notation $\mathcal{N}(y; x, \sigma^2)$ is defined as $1/(\sigma\sqrt{2\pi})\exp(-(y-x)^2/(2\sigma^2))$. Also, knowing the parameter and noise pdf, or equivalently $p_{Y|X}(y|x)$ and $p(x)$, the function $f(y)$ can be numerically calculated using the following formula

$$\begin{aligned} f(y) = E\{X|y\} &= \int_x x\ p_{X|Y}(x|y)dx \\ &= \int x\ \frac{p_{Y|X}(y|x)p(x)}{\int_x p_{Y|X}(y|x)p(x)dx}dx. \end{aligned} \tag{B.4}$$

Having $f(y)$, its inverse and derivative, can also be numerically calculated and used in (B.3) to find $p_{\hat{X}|X}(\hat{x}|x)$. Finally, D_1 can be found from (B.2).

The second term in (B.1), i.e., D_2, is the distortion function for quantizing the random variable $\hat{X}$ with a rate R. A lower bound for D_2 can be found as $\tilde{D}_2 = Q_1 2^{-2R}$, where Q_1 is found from the following equality [108]

$$h(p_{\hat{X}}(\hat{x})) = \frac{1}{2}\log_2(2\pi e Q_1), \tag{B.5}$$

where $h(p_{\hat{X}}(\hat{x}))$ is the entropy of the random variable $\hat{X}$. To calculate the entropy, first the pdf $p_{\hat{X}}(\hat{x})$ is calculated using a similar approach as in (B.3). Finally, having calculated D_1 and $\tilde{D}_2$, the approximate lower bound D is found to compare to the performance of our distributed estimation-quantization method.

B.2 Unbiasedness of the Estimator

The amount of bias for an estimator is defined as

$$E\{X - \hat{X}\} = E\{X\} - E\{\hat{X}\}, \tag{B.6}$$

where X is the unknown random parameter and $\hat{X}$ is its estimated value. In our method, X is going to be estimated from N binary data using a MMSE estimator

in the FC, i.e., $\hat{X} = E\{X|z_1, \cdots, z_N\}$. Therefore,

$$\begin{aligned} E\{\hat{X}\} &= E\{E\{X|z_1, \cdots, z_N\}\} \\ &= \sum_{z_1,\cdots,z_N} E\{X|z_1, \cdots, z_N\} p(z_1, \cdots, z_N) \\ &= \sum_{z_1,\cdots,z_N} \left(\int_X X \; p(X|z_1, \cdots, z_N) dX \right) p(z_1, \cdots, z_N) \\ &= \int_X X \sum_{z_1,\cdots,z_N} p(X, z_1, \cdots, z_N) dX \\ &= \int_X X p(X) dX = E\{X\}. \end{aligned} \tag{B.7}$$

The last equality ensures the unbiasedness of the method, when an MMSE estimator is used in the FC.

B.3 CRLB

The CRLB for estimating an unknown random variable X from a noisy measurement Z is the inverse of the Fisher information metric, which can be obtained as

$$J = E\Big\{ \big[\frac{\partial \ln p(Z, X)}{\partial X} \big]^2 \Big\} \tag{B.8}$$

where $p(Z, X)$ is the joint probability distribution of X and Z, and E indicates the expected value with respect to both X and Z. It can be proved that the above definition is identical to [59]

$$J = -E\Big\{ \big[\frac{\partial^2 \ln p(Z, X)}{\partial X^2} \big] \Big\}. \tag{B.9}$$

Having $p(Z, X) = p(Z|X)p(X)$, (B.9) can be written as

$$J = \underbrace{-E\Big\{ \big[\frac{\partial^2 \ln p(Z|X)}{\partial X^2} \big] \Big\}}_{J_1} \quad \underbrace{-E\Big\{ \big[\frac{\partial^2 \ln p(X)}{\partial X^2} \big] \Big\}}_{J_2}, \tag{B.10}$$

where the first term J_1 is related to the likelihood of measurements and the second term J_2 depends only on the distribution of X.

Now, suppose that there are N binary measurements available from X. In other words, $Z = [z_1, \cdots, z_N]$. The likelihood function $p(Z|X)$ can be written as [109]

$$\begin{aligned} p(z_1, \cdots, z_N|X) &= \prod_{n=1}^{N} p(z_n|X) \\ &= \prod_{n=1}^{N} \mathrm{P1}_n(X)^{(z_n)} \mathrm{P0}_n(X)^{(1-z_n)}, \end{aligned} \tag{B.11}$$

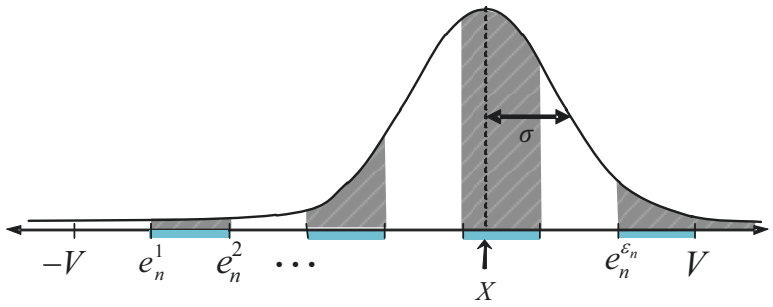

Figure B.1: The probability $p(z_n=1|X)$.

where $\mathrm{P1}_n(X)$ is the probability that the nth bit is 1 when the parameter value is X, i.e., $p(z_n=1|X)$. Similarly, $\mathrm{P0}_n(X)=p(z_n=0|X)$. If the nth local binary quantization pattern is defined by the set of cell edges $\{e_n^1, e_n^2, \cdots, e_n^{\epsilon_n}\}$ mapping the first cell to 0, for Gaussian measurement noise with variance σ, $\mathrm{P1}_n(X)$ can be written as (see Fig. B.1).

$$\mathrm{P1}_n(X)=\sum_{i=1}^{\epsilon_n}(-1)^{i+1}Q(\frac{e_n^i-X}{\sigma}). \tag{B.12}$$

Now, $\ln p(Z|X)$ can be written as

$$\ln p(Z|X)=\sum_{n=1}^{N}\Big((z_n)\ln \mathrm{P1}_n(X)+(1-z_n)\ln \mathrm{P0}_n(X)\Big). \tag{B.13}$$

Therefore, its first and second derivative can be calculated as

$$\frac{\partial \ln p(Z|X)}{\partial X}=\sum_{n=1}^{N}\Big\{(z_n)\frac{\frac{\partial \mathrm{P1}_n(X)}{\partial X}}{\mathrm{P1}_n(X)}+(1-z_n)\frac{\frac{\partial \mathrm{P0}_n(X)}{\partial X}}{\mathrm{P0}_n(X)}\Big\} \tag{B.14}$$

and

$$\begin{aligned}\frac{\partial^2 \ln p(Z|X)}{\partial X^2}=&\\ \sum_{n=1}^{N}\Big\{(z_n)\{&\frac{\frac{\partial^2 \mathrm{P1}_n(X)}{\partial X^2}}{\mathrm{P1}_n(X)}-(\frac{\frac{\partial \mathrm{P1}_n(X)}{\partial X}}{\mathrm{P1}_n(X)})^2\}\\ &+(1-z_n)\{\frac{\frac{\partial^2 \mathrm{P0}_n(X)}{\partial X^2}}{\mathrm{P0}_n(X)}-(\frac{\frac{\partial \mathrm{P0}_n(X)}{\partial X}}{\mathrm{P0}_n(X)})^2\}\Big\}.\end{aligned} \tag{B.15}$$

To find the first term, J_1, in (B.10), the expectation of the second derivation of $p(Z|X)$, i.e., (B.15), is calculated as

$$\begin{aligned}J1=&-E\{[\frac{\partial^2 \ln p(Z|X)}{\partial X^2}]\}=\\ &-\int_X\sum_Z[\frac{\partial^2 \ln p(Z|X)}{\partial X^2}]p(Z|X)p(X)dX.\end{aligned} \tag{B.16}$$

Inserting (B.15) in (B.16), and calculating the integral over Z, we have

$$J1 = -\sum_{n=1}^{N} \int_X \left(\frac{\partial^2 \mathrm{P1}_n(X)}{\partial X^2} - \frac{\left(\frac{\partial \mathrm{P1}_n(X)}{\partial X}\right)^2}{\mathrm{P1}_n(X)}\right) p(X) dX \tag{B.17}$$
$$-\sum_{n=1}^{N} \int_X \left(\frac{\partial^2 \mathrm{P0}_n(X)}{\partial X^2} - \frac{\left(\frac{\partial \mathrm{P0}_n(X)}{\partial X}\right)^2}{\mathrm{P0}_n(X)}\right) p(X) dX.$$

To further simplify the above, note that for any binary local binary quantization patterns,

$$\mathrm{P0}_n(X) = 1 - \mathrm{P1}_n(X),$$

hence,

$$\frac{\partial \mathrm{P0}_n(X)}{\partial X} = -\frac{\partial \mathrm{P1}_n(X)}{\partial X} \qquad \frac{\partial^2 \mathrm{P0}_n(X)}{\partial X^2} = -\frac{\partial^2 \mathrm{P1}_n(X)}{\partial X^2} \tag{B.18}$$

Therefore, (B.17) is reduced to

$$J1 = \sum_{n=1}^{N} \int_X \left\{\frac{\left(\frac{\partial \mathrm{P1}_n(X)}{\partial X}\right)^2}{\mathrm{P1}_n(X)} + \frac{\left(\frac{\partial \mathrm{P1}_n(X)}{\partial X}\right)^2}{1 - \mathrm{P1}_n(X)}\right\} p(X) dX. \tag{B.19}$$

For a random variable with uniform distribution, the second term, J_2 in (B.10) is zero; and therefore, the CRLB reduces to $1/J_1$. For a Gaussian random variable, J_2 can be easily calculated as

$$J_2 = -E\left\{\left[\frac{\partial^2 \ln \mathcal{N}(x, \sigma_x^2)}{\partial X^2}\right]\right\} = \frac{1}{\sigma_x^2}, \tag{B.20}$$

and the CRLB will be $1/(J_1 + J_2)$.

Appendix C

Details for Chapter 6

C.1 Conditional Bit-Error Probability

The probability that bit b is wrong conditioned on X, i.e., $p(\alpha_b|X)$, is written as

$$p(\alpha_b|X) = \begin{cases} \mathrm{P1}_b(X) & X \in U_0 \\ 1 - \mathrm{P1}_b(X) & X \in U_1 \end{cases}$$

where $\mathrm{P1}_b(X)$ is the probability that bit b is quantized to 1 conditioned on X. Also, U_1 is the union of all divisions in the quantization rule $\mathcal{P}_b$ that map to 1. And we have $U_0 \cup U_1 = [-1, 1]$. Defining $\mathcal{P}_b$ by the set of edges $\{\gamma_1, \gamma_2, \cdots\}$, and assuming sensor n_b has Gaussian measurement noise with variance σ, $\mathrm{P1}_b(X)$ can be written as

$$\mathrm{P1}_b(X) = \sum_i (-1)^{i+1} Q(\frac{\gamma_i - X}{\sigma}), \tag{C.1}$$

where $Q(\cdot)$ indicates the Q-function. In Fig. C.1, we have shown $p(\alpha_b|X)$; $1 \leq b \leq 4$, for Gray labeling scheme, where all sensors have Gaussian measurement noise with variance $\sigma^2 = 0.03$.

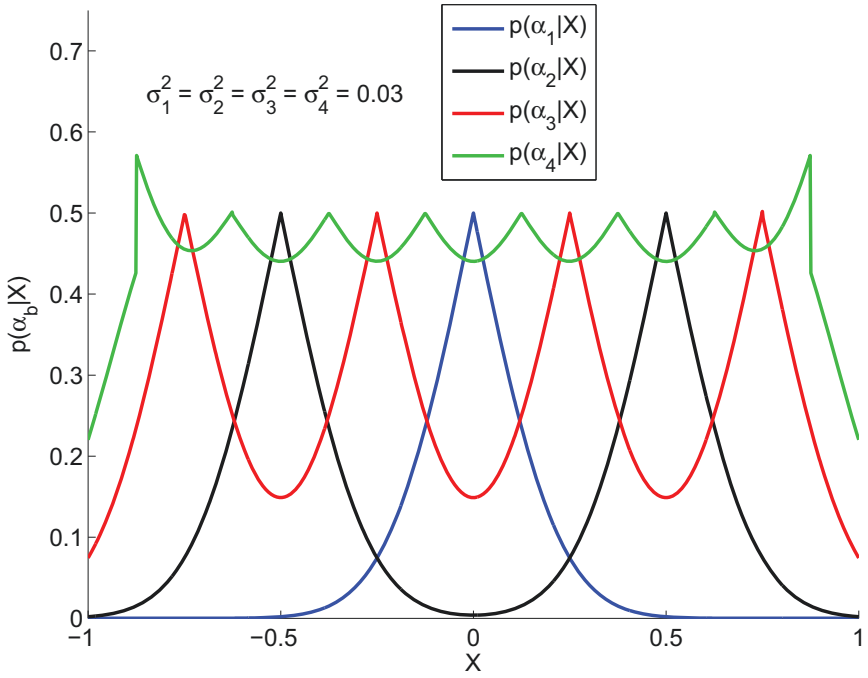

Figure C.1: Conditional bit error probabilities $p(\alpha_b|X)$ for the first 4 local binary quantization patterns of Gray labeling.