Advanced Topics in Estimation and Information Theory

ADVANCED TOPICS IN ESTIMATION AND INFORMATION THEORY

BY

AMIN ZIA

SEPTEMBER 2007

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2007) McMaster University

(Electrical & Computer Engineering) Hamilton, Ontario

TITLE:	Advanced Topics in Estimation and Information Theory
AUTHOR:	Amin Zia M.Sc. (Electrical Engineering) Iran University of Science and Technology, Tehran, Iran
SUPERVISOR:	Drs. James P. Reilly, Shahram Shirani
NUMBER OF PAGES:	xxx, 239

Dedicated to my mother and father,

and my lovely wife

Abstract

The main theme of this dissertation is statistical estimation and information theory. There are three related topics including "distributed estimation", "an information geometric approach to ML estimation with incomplete data" and "joint identification and estimation in non–linear state space using Bayesian filters". The expectation–maximization (EM) algorithm, as an iterative estimation technique for dealing with incomplete data is the common bond that binds these three topics together.

1. *Distributed estimation*

   Distributed estimation involves the study of estimation theory in an information theoretic framework. This field concerns the following question: "What if the purpose of communications in a distributed environment is parameter estimation rather than source reconstruction?" The first part of this thesis is dedicated to designing low–complexity iterative algorithms for distributed estimation. The algorithm design, in this case, involves transmission of statistics via communication systems. Therefore, the first question raised is "whether the code rates in distributed estimation are different from those in conventional communications?" Surprisingly, under certain conditions, the answer is found to be negative. It is shown that for fixed parameters, the achievable rates coincide with rates in conventional distributed coding of correlated sources (i.e. Slepian–Wolf region). In order to prove the main theorem, we also devise a novel

distributed binning scheme and a new theorem in *Large deviation theory* that are used for proving our distributed coding theorem. The proof of the converse is implemented by a generalized *Fano's inequality* for distributed estimation.

Determination of the region of achievable rates for efficient estimation of a general source is an extremely difficult problem. This fact is the motivation for proving a theorem that provides a method for determining the region of achievable rates for a large class of sources with a convex mutual information with respect to the unknown parameters.

With a given set of rates, an efficient implementation of universal coding schemes for distributed estimation based on the expectation maximization (EM) technique is presented. Since the correlation channel between the sources is assumed to be unknown at the joint decoder, previously proposed distributed coding schemes are not useful for this purpose. Therefore, LDPC-based coset–coding schemes are extended to the case where the correlation channel is unknown at the decoder. The basic idea is to implement a low–complexity version of the EM algorithm on a factor–graph that includes an LDPC decoding mechanism.

2. *Information geometric approach to ML estimation with incomplete data*

The stochastic maximum likelihood estimation of parameters with incomplete data is cast in an information geometric framework. In this vein we develop the *information geometric identification (IGID)* algorithm, that provides an alternative iterative solution to the incomplete–data estimation problem. The algorithm consists of iterative alternating projections on two sets of probability distributions (PD); i.e., likelihood PD's and data empirical distributions. A Gaussian assumption on the source distribution permits a closed form low–complexity solution for these projections. The method is applicable to a wide

range of problems; however the emphasis is on semi–blind identification of unknown parameters in a multi-input multi-output (MIMO) communications system.

3. *Joint identification and estimation in non–linear state space using Bayesian filters*

   There are situations in estimation where nonlinear state–space models where the model parameters or the model structure itself are not known a priori or are known only partially. In these scenarios, standard estimation algorithms like the extended Kalman Filter (EKF), which assume perfect knowledge of the model parameters, are not accurate. The nonlinear state estimation problem with possibly non-Gaussian noise in the presence of measurement model uncertainty is modeled as a special case of maximum likelihood estimation with incomplete data. The EM algorithm is used to solve the problem. The expectation (E) step is implemented by a particle filter that is initialized by a Monte-Carlo Markov chain algorithm. In the maximization (M) step, a nonlinear regression method, here using a mixture of Gaussians (MoG), is used to approximate (identify) the uncertain model equations. The proposed procedure is used to solve a highly nonlinear bearing-only tracking problem, as well as the sensor registration problem in a *multi–sensor fusion* scenario.

Acknowledgements

The last couple of years have been very challenging and enjoyable; I had the opportunity to explore a wide range of interesting subjects which I had no prior knowledge and for which the degree of risk was quite high.

First of all, I would like to thank Dr. Jim Reilly for supervising me during this challenge and for his continues encouragements. His enthusiasm, approachability and patience were essential to the completion of this dissertation. I have no doubt that without his exceptional support and insight this job would not have been completed. I am also very grateful to Dr. Shahram Shirani, my co–supervisor, for his patience during many hours of discussions. I appreciate his continuous support in different stages of my studies. I learned a great deal about how to approach a problem in discussions I had with Dr. Tim Field. I would like to offer appreciation for invaluable discussions we had on information geometry. I would also like to thank Dr. Thia Kirubarajan, Dr. Tim Davidson, Dr. Steve Hrinalovic and Dr. John Wilson for their most insightful comments and discussions. I would like also to acknowledge joint work with Dr. Kris Huber, during which I learned a lot about MIMO diversity codes and found the chance to work with real-world MIMO communication channel measurements. Part of the simulations in Chapter 5 were performed by my colleagues Derek Yee and Kumaradevan Punithakumar whose help is greatly appreciated.

Further thanks are also due to the ECE department and staff. In particular I would like to thank Cheryl Gies and Helen Jachna who patiently solved all of my

non-research related problems patiently - what would this department do without you?.

This period of my life would have not started without inspirations of my dear family. I would like to deeply thank my mother and father for their endless and unconditional love and support, my brother for his life-long encouragement, and my sisters for their exceptional wisdom and love. I was so lucky to find my life partner at a critical time of my life. I would like to deeply thank my most wonderful wife who supported me through the most difficult moments of this period with her exceptional encouragement, patience, and love.

To all you involved in this chapter of my life - THANK YOU!

Acronyms

AB	Ahlswede and Burnashev
AEP	The asymptotic equipartition property
APP	A posterior probability
BER	Bit-error-rate
BSC	Binary symmetric channel
BSS	Binary symmetric source
CRLB	Cramer-Rao lower bound
DISCUS	Distributed source coding using syndrome
ECEF	Earth centered earth fixed
EKF	Extended Kalman filter
EM	Expectation maximization
FI	Fisher information
FIM	Fisher information matrix
GSBS	Generalized pseudo-Bayesian estimator
HA	Han–Amari
IGID	Information geometric identification
i.i.d.	independent and identically distributed
IMM	Interactive multiple model estimation
ISI	Inter–symbol interference

KL	Kullback-Liebler distance
LDPC	Low–density parity check
LLR	Log–likelihood ratio
MCMC	Markov-Chain Monte-Carlo
MH	Metropolis-Hastings
MIMO	Multi-input multi-output
ML	Maximum likelihood
MLE	Maximum likelihood estimation
MM	Multiple model
MoG	Mixture of Gaussians
MPA	Message–passing algorithm
MSE	Mean square error
OFDM	Orthogonal–frequency division multiplex
PCRLB	The posterior Cramér-Rao lower bound
pdf	Probability density function
PD	Probability distribution
PF	Particle filter
QAM	Quadrature amplitude modulation
QPSK	Quadrature phase shift keying
RV	Random variable
SER	Symbol-error-rate
SI	Side–information
SNR	Signal-to-noise ratio
SW	Slepian–Wolf
ZB	Zhang–Berger

Notation

Symbol	Definition
X	Random variable
x	A realization of a random variable
$\boldsymbol{C}$	Matrix
$\boldsymbol{C}^T$	Matrix transpose
$\boldsymbol{C}'$	Hermitian transpose
$\lvert\boldsymbol{C}\rvert$	Determinant
$\lvert\mathcal{A}\rvert$	Cardinality of a set
$\lvert a\rvert$	Absolute value for a scaler
$\mathrm{Tr}(\boldsymbol{Y})$	Trace of a matrix
E	Expectation operator
$\Pr(A)$	Probability of the event A
$p(X)$ or $p(X^n)$	Probability density function
$x \sim p(X)$	x is distributed or drawn from $p(X)$
$p(X;\theta)$	Probability density of x parameterized with θ
Information theory (Chapters 2-3)	
X^n	Vector of random variables $X_1, ...X_n$
x^n	A Realization of a vector of random variables
X_i	Element i of a vector

x^j	Abbreviated reference to the j^{th} n-vector, i.e. x^{nj}
$H(X)$	Entropy of X
$H_b(X)$	Binary entropy
$H(X,Y)$	Joint entropy
$I(X,Y)$	Mutual information
$H(X\|Y)$	Conditional entropy of X given Y
$N(a\|x^n)$	Relative frequency of a in sequence x^n
$\tilde{P}_{x^n}$	The type of a sequence x^n
$\tilde{P}_{x^n y^n}$	The joint–type of a pair (x^n, y^n)
$\mathcal{P}_n$	Set of types with denominator n
$T(P_x)$	The class of a type
$H(P_{x^n})$	Entropy of a type
$d_H(x^n, y^n)$	Hamming distance
Channel estimation (Chapter 4)	
$\boldsymbol{Q}$	A matrix
$\boldsymbol{x}$	A vector
$\mathcal{Q}$	Set of probability distributions
$N(\boldsymbol{\mu}, \boldsymbol{\Psi})$	A multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Psi}$
$q(\boldsymbol{x}, \boldsymbol{y})$	Joint distribution of two vectors
$q(\boldsymbol{x})$	Marginal distribution of a vector
$q(\boldsymbol{x}\|\boldsymbol{y})$	Conditional distribution of $\boldsymbol{x}$ given $\boldsymbol{y}$
t	The IGID algorithm iteration index
k	Temporal index
Nonlinear filtering (Chapter 5)	
$\boldsymbol{z}(t)$	A vector as a function of time

$\boldsymbol{z}$	The set of all values of $\boldsymbol{z}(t)$ over the temporal range
$\boldsymbol{A}$	Matrix $\boldsymbol{A}$
L	The number of data points
t	Discrete time index ($t = 1, \ldots, L$)
k	The EM iteration index ($k = 1, 2 \ldots$)
$k = 1$	The initialization step in the EM algorithm
i	The particle index, ($i = 1, \ldots, N$)
N	Number of particles
$\mathcal{N}(\boldsymbol{m}, \boldsymbol{\Sigma})$	A Gaussian distribution with mean $\boldsymbol{m}$ and covariance $\boldsymbol{\Sigma}$

Contents

Abstract **v**

Acknowledgements **ix**

Acronyms **xi**

Notation **xiii**

1 Introduction **1**

1.1 Summary 1

1.1.1 Preliminary definitions from statistics: 4

1.1.1.1 Fisher information 4

1.1.1.2 Additivity of Fisher information 5

1.1.1.3 Sufficient Statistics 6

1.1.1.4 Maximum likelihood (ML) parameter estimation . . 7

1.1.1.5 Estimator unbiasedness and efficiency 8

1.1.1.6 Estimator asymptotic consistency and efficiency . . . 9

1.1.1.7 Consistency and efficiency of ML estimation 10

1.1.1.8 Loss in Fisher information 11

1.1.2 ML estimation using incomplete data: the EM algorithm: . . . 11

1.2 Communications for estimation 13

1.2.1 Motivation: A distributed weather network 14
1.3 Problem Statement 16
1.3.1 A Brief Review of Literature 17
1.3.2 Thesis contribution: 19
1.3.3 Relationship with Distributed Coding 21
1.3.4 Relation to network information theory 21
1.4 Information geometric approach to MLE 22
1.4.1 Motivation 22
1.4.2 Problem statement 22
1.4.3 The EM-based channel estimation algorithms 24
1.4.4 Thesis contribution 25
1.5 Joint identification and estimation using Bayesian filters and MCMC 26
1.5.1 Motivation 26
1.5.2 Problem statement 26
1.6 Contribution of the thesis 27

2 On Multiterminal Estimation Rates **29**
2.1 Introduction 29
2.1.1 A Brief Review of Literature 30
2.1.2 Thesis contribution: 32
2.2 Problem Statement 35
2.3 Information theory- some definitions and theorems 36
2.3.1 A quick review of fundamentals 38
2.3.1.1 Typical and jointly typical sets [24] 38
2.3.1.2 Marginal and joint types 40
2.3.1.3 Set of types ($\mathcal{P}_n$) 40
2.3.1.4 Type class 41

2.3.1.5 Entropy of a type 41
2.3.2 Large deviation theory 42
2.4 Sufficiency of the joint–type 46
2.5 Coding for distributed estimation 48
2.5.1 Coding scheme 49
2.6 Proof of the converse 55
2.6.1 Proof of the converse to Theorem 2.5.1 (Necessary condition) . 58
2.7 On the region of achievable rates 59
2.8 Discussion 63
2.8.1 A note on the method of Han and Amari [47]: 64
2.8.2 A note on "Modulo-two adder" source network: 68

3 Distributed Parameter Estimation **73**
3.1 Introduction 73
3.2 Distributed ML Estimation with Side Information 77
3.2.1 Binary symmetric source 77
3.2.2 Problem Statement 78
3.2.3 Binary symmetric channel model and MLE 79
3.2.3.1 MLE with complete data 80
3.2.3.2 The MLE with compressed data using side–information 81
3.3 Linear Block Parity Check Codes 83
3.3.1 Example– Hamming $(7, 4)$ channel code 84
3.3.2 Maximum likelihood detection 85
3.3.3 Syndrome and coset 85
3.3.4 Syndrome decoding 87
3.4 Linear Block Codes for Distributed Coding 88
3.4.1 Example: The Hamming $(7, 4)$ distributed source code 89

3.4.2 Distributed source coding using syndrome (DISCUS) 91
3.5 Distributed estimation using EM 93
3.5.1 Encoder 93
3.5.2 Decoder/Estimator 93
3.5.3 Expectation step (E-step) 94
3.5.4 Maximization step (M–step) 95
3.6 Low density parity check codes 96
3.6.1 LDPC Codes– Decoding Algorithm: 98
3.6.2 Messages over the factor graph (APP case) 101
3.6.2.1 Messages from the variable nodes to the check nodes 102
3.6.2.2 Messages from the check nodes to the variable nodes 102
3.6.2.3 Initialization 103
3.6.3 Summary of MPA (APP Case:) 103
3.6.4 Messages over the factor graph (LLR case) 104
3.6.4.1 Messages from variable nodes to check nodes 104
3.6.4.2 Messages from check nodes to variable nodes 105
3.6.4.3 Initialization 106
3.6.5 Summary of MPA (LLR Case:) 106
3.7 Low-complexity distributed estimation 107
3.7.1 LDPC–based syndrome decoding for distributed coding 109
3.7.1.1 Messages from check nodes to variable nodes- APP case110
3.7.1.2 Messages from check nodes to variable nodes- LLR case111
3.7.2 The E-Step 111
3.7.3 The M-Step 114
3.7.4 Scheduling the Decoding/Estimation 115
3.7.5 The algorithm 115
3.8 Region of Achievable Rates 117

3.9 Simulations 119
3.10 Discussion 126

4 An Information Geometric Approach to MLE **129**
4.1 Introduction 129
4.2 Stochastic ML Estimation 132
4.2.1 The Information Geometric Approach to Stochastic ML Estimation 134
4.3 Application to Semi–Blind Channel Identification 138
4.3.1 Signal Distributions 138
4.3.2 The First Projection: Computing the Best Complete-Data Distribution 139
4.3.3 The Second Projection: the Complete-Data ML Estimation . . 140
4.3.4 Initialization Using Training 142
4.3.5 The IGID Algorithm: Summary 143
4.3.6 Convergence of the IGID Algorithm 143
4.4 Simulations 144
4.4.1 Channel Estimation 144
4.4.2 Symbol-Error-Rate (SER) 146
4.4.3 Discussion 151
4.5 Conclusions 153

5 An EM Algorithm for State Estimation **155**
5.1 Introduction 155
5.2 Nonlinear State Estimation using EM 159
5.3 The EM–PF Algorithm 162
5.3.1 The E-step: Estimation of States by the Particle Filter 162
5.3.2 The M-Step 168

5.3.3 Summary 171
5.4 Bearing-only Tracking 173
5.4.1 Problem Statement 173
5.4.2 Simulation Results 175
5.5 Sensor Registration 181
5.5.1 Problem Statement 183
5.5.2 Simulation Results 187
5.6 Conclusions 188

6 Conclusions **191**
6.1 Conclusions 191
6.1.1 Joint identification and estimation using Bayesian filters and MCMC 191
6.1.2 Wireless MIMO blind channel estimation 192
6.1.3 Distributed estimation 193
6.2 Suggestions for further investigations 194
6.2.1 Estimation rate–distortion Theory: 194
6.2.2 (Network) Information theory and statistical analysis 195

A Proof of Theorem (1.1.5) **197**

B Proof of Lemma (2.3.8, part (a)) **203**

C Proof of Theorem 3.6.2 **207**

D IGID and the Variational EM algorithm **209**
D.1 The E-Step vs. the First Projection 211
D.2 The M-Step vs. the Second Projection 212

E EM-based Blind Identification Algorithms **213**

F Proof of (4.23) **215**

G Proof of (4.30) **217**

H Proof of (4.31) and (4.32) **219**

I Proof of (4.36) **221**

J Proof of (5.28) and (5.29) **223**

J.1 Solution for θ_k 223

J.2 Solution for Q 224

List of Tables

3.1 The binary entropy of binary symmetric source $BSS(\rho)$ for different values of parameter ρ. 118

5.1 Parameters for the bearing–only tracking simulation example. 176

List of Figures

1.1 Distributed estimation 17

2.1 Codebook for a distributed binning scheme 50

2.2 A convex mutual information function with minimum at θ_m. The mutual information for any parameter $\theta_s \leq \theta_m$ is more than I_m corresponding to θ_m. 61

2.3 Region of rates corresponding to parameters θ_m and θ_s in Figure 2.2 . 61

2.4 The mutual information function of BSS 63

3.1 Distributed estimation with side–information 78

3.2 A bipartite graph for the LDPC codes 97

3.3 (a) EM factor graph and corresponding messages, (b) EM-LDPC Factor Graph for ML estimation with side–information 109

3.4 L = 6 iterations of the EM algorithm for estimation of $\rho = 0.1$ with random initial value (shown here for $\rho_0 = 0.4$, $n = 200$, $R_x \in [0.5, 1.0]$ 118

3.5 (a) L = 6 iterations of the EM algorithm for estimation of $\rho = 0.10$ with random initial value, $n = 200$, $R_x \in [0.5, 1.0]$, and (b) MSE for L = 6 iterations of the EM algorithm, parameters similar to part (a) . . 121

3.6 (a) L = 6 iterations of the EM algorithm for estimation of $\rho = 0.15$ with random initial value, $n = 200$, $R_x \in [0.5, 1.0]$, and (b) MSE for L = 6 iterations of the EM algorithm, parameters similar to part (a) . . 121

3.7 (a) L = 6 iterations of the EM algorithm for estimation of $\rho = 0.20$ with random initial value (shown here for $\rho_0 = 0.4$, $n = 200$, $R_x \in [0.5, 1.0]$, and (b) MSE for L = 6 iterations of the EM algorithm, parameters similar to part (a) 122

3.8 (a) L = 6 iterations of the EM algorithm for estimation of $\rho = 0.25$ with random initial value, $n = 200$, $R_x \in [0.5, 1.0]$, and (b) MSE for L = 6 iterations of the EM algorithm, parameters similar to part (a) . . 122

3.9 (a) MSE for estimation of $\rho = 0.10$ after convergence of the EM algorithm for different available rates $R_x \in [0.5, 1.0]$ compared with the attainable CRLB (Eq. 3.54) (b) Similar to part (a) for $\rho = 0.15$ . . . 123

3.10 (a) MSE for estimation of $\rho = 0.20$ after convergence of the EM algorithm for different available rates $R_x \in [0.5, 1.0]$ compared with attainable CRLB (Eq. 3.54) (b) Similar to part (a) for $\rho = 0.25$ 123

3.11 (a) Estimation of $\rho = 0.10$ versus the EM iterations for different code lengths: Here $R_x = 0.65$ (b) MSE after convergence for different code lengths 124

3.12 (a) Estimation of $\rho = 0.20$ versus the EM iterations for different code lengths, here $R_x = 0.65$ (b) MSE after convergence for different code lengths 124

3.13 (a) Estimation of $\rho = 0.20$ versus the EM iterations for different code lengths, here $R_x = 0.90$ (b) MSE after convergence for different code lengths 125

3.14 (a) BER vs. available rates $R_x \in [0.5, 1.0]$ for different steps of the EM algorithm, here $\rho = 0.10$, and (b) Similar to part (b) for here $\rho = 0.15$ 125

3.15 (a) (a) BER vs. available rates $R_x \in [0.5, 1.0]$ for different steps of the EM algorithm, here $\rho = 0.20$, and (b) Similar to part (b) for here $\rho = 0.25$ 126

4.1 Convergence (*rms* error vs. iteration index) for channel gain estimation in the low-SNR regime (SNR=6dB) in a 2 by 2 MIMO communication system with 16-QAM modulation. "ML with Training" uses the whole block of data as a training sequence, whereas the EM and the IGID algorithms each use 10% of the data block for training. The error is evaluated over 50 Monte Carlo runs. 146

4.2 Same as Figure 4.1, except SNR = 16 dB 147

4.3 Symbol Error Rate (SER) curves for QPSK modulation for a block length of $L = 100$. 148

4.4 Same as Figure 4.3, except the block length $L = 1000$. 148

4.5 SER curves for 16-QAM modulation for a block length of $L = 100$. . . 149

4.6 SER curves for 16-QAM modulation for a block length of $L = 1000$. . 149

4.7 SER curves for 64-QAM modulation for a block length of $L = 1000$. . 150

4.8 $D(p_t||q_t)$ vs. iteration index t. 151

5.1 Block diagram of the EM–PF algorithm, which gives state and model parameter estimates, over the block $t = 1, \ldots, L$. 163

5.2 Unbiased, noise-free but perturbed measurements obtained from (5.36) with $\beta = 0$ (bottom), and the biased, perturbed, noisy measurements from (5.38), that are input to the algorithm (top). 176

5.3 Position (a) and velocity (b) tracking trajectories for the EM-PF algorithm over four successive iterations. 178

5.4 Root MSE of the position (a) and velocity (b) state estimates vs. time over 50 Monte–Carlo runs of the EM-PF algorithm for four iterations. 178

5.5 Root MSE of the position (a) and velocity (b) state estimates vs. time of the EM-PF algorithm at the fourth iteration, along with the corresponding PCRLB curves. 179

5.6 Root MSE vs. observation noise variance for the bearing–only tracking problem. 180
5.7 Position and velocity estimates for the IMM-EKF algorithm, applied to the bearing-only tracking problem. 181
5.8 Position and velocity estimate RMSEs for the IMM-EKF algorithm, applied to the bearing-only tracking problem over the time interval 0 to 40s. 182
5.9 Position and velocity estimate RMSEs for the IMM-EKF algorithm, applied to the bearing-only tracking problem, for the time interval from 20s to 40s. 183
5.10 True (circle) and biased (dot) target trajectories estimated by two sensors (star) 188
5.11 (a)True and registered target trajectories after application of the EM-PF algorithm, and (b) RMS position error for two sensors vs. iteration number 189

Chapter 1

Introduction

1.1 Summary

The main theme of this dissertation is statistical estimation. There are three related topics including "distributed estimation", "an information geometric approach to ML estimation with incomplete data" and "joint identification and estimation in non–linear state space using Bayesian filters". The expectation–maximization (EM) algorithm, as an iterative estimation technique for dealing with incomplete data is the common bond that binds these three topics together.

Distributed estimation: Distributed estimation involves the study of estimation theory in an information theoretic framework. This field concerns the question: "What if the purpose of communications in a distributed environment is parameter estimation rather than source reconstruction?" The first part of this thesis is dedicated to designing low–complexity iterative algorithms for distributed estimation. The algorithm design, in this case, involves transmission of statistics via communication systems. Therefore, the first question raised in the communications engineer's mind is the extent that communication resources are sufficient (and necessary) for such a purpose. Therefore, in order to answer these questions, as well as to provide a

proper understanding of the problem, one needs to first determine the sufficient (and necessary) communications resources (rates) for efficient distributed estimation.

Chapter 2 is the result of an effort to answer the question: "Whether the code rates in distributed estimation are different from those in conventional communications?" Surprisingly, under certain conditions, the answer is found to be negative. It is shown that for fixed parameters, the achievable rates coincide with rates in conventional distributed coding of correlated sources (i.e. the Slepian–Wolf region). A novel theorem for bounding the achievable rates for a large class of sources is also presented.

With a given set of rates, in Chapter 3, an efficient implementation of universal coding schemes for distributed estimation based on the expectation maximization (EM) technique is presented. Since the correlation channel between the sources is assumed to be unknown at the joint decoder, previously proposed distributed coding schemes are not useful for this purpose. Therefore, LDPC-based coset–coding schemes are extended to the case where the correlation channel is unknown at the decoder. The basic idea is to implement a low–complexity version of the EM algorithm on a factor–graph that includes an LDPC decoding mechanism.

Information geometric approach to ML estimation: Here, the stochastic maximum likelihood estimation of parameters with incomplete data is cast in an information geometric framework. In this vein we develop the *information geometric identification (IGID)* algorithm, that provides an iterative alternative solution to the incomplete–data estimation problem. The algorithm consists of iterative alternating projections on two sets of probability distributions (PD); i.e., likelihood PD's and data empirical distributions. A Gaussian assumption on the source distribution permits a closed form low–complexity solution for these projections. The method is applicable to a wide range of problems; however, in this chapter the emphasis is on semi–blind identification of unknown parameters in a multi-input multi-output (MIMO) communications

system.

Joint identification and estimation in non–linear state space using Bayesian filters: There are situations in estimation with nonlinear state–space models where the model parameters or the model structure itself are not known a priori or are known only partially. In these scenarios, standard estimation algorithms like the extended Kalman Filter (EKF), which assume perfect knowledge of the model parameters, are not accurate. The nonlinear state estimation problem with possibly non-Gaussian noise in the presence of measurement model uncertainty is modeled as a special case of maximum likelihood estimation with incomplete data. The EM algorithm is used to solve the problem. The expectation (E) step is implemented by a particle filter that is initialized by a Monte-Carlo Markov chain algorithm. In the maximization (M) step a nonlinear regression method using a mixture of Gaussians (MoG) is used to approximate (identify) the uncertain model equations. The proposed procedure is used to solve a highly nonlinear bearing-only tracking problem, as well as the sensor registration problem in a *multi–sensor fusion* scenario.

All proposed algorithms in this dissertation are involved, in one way or another, with maximum likelihood estimation using incomplete data and the application of the expectation–maximization (EM) algorithm [35]. This algorithm is the most renowned technique in estimation with incomplete data and the main body of most iterative estimation algorithms currently used in engineering. Therefore, from the algorithmic viewpoint, one of the main themes in this dissertation is modeling the problems at hand into an *incomplete–data* problem, and then solving them by a variant of the EM algorithm.

Organization of chapter: This chapter begins with a brief review of some of important fundamentals in statistics and a tutorial on the maximum likelihood estimation using incomplete data and the EM algorithm.

Then, the three related but different problems that are the subject of the following

chapters are introduced and the contributions of the thesis are reviewed.

Notation: In this dissertation, a random variable and its particular realization are represented by uppercase and lowercase letters, X and x, respectively. A sequence (vector) of n random variables and any corresponding realization are shown by superscript n, e.g. X^n and x^n, respectively. When a reference to any element of a sequence (vector) is needed a proper subscript is used, e.g. X_i and x_i for element i of a random sequence and its realization, respectively. When more than a reference to a vector is needed, where clear from context, the superscript n is omitted.

1.1.1 Preliminary definitions from statistics:

Suppose $\mathcal{X}$ and $\mathcal{Y}$ are sets with finite or countably infinite cardinality. The joint probability distribution $Q(X,Y;\theta)$ is a function $Q : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{R}$ which satisfies:

$$Q(X,Y) \geq 0 \quad (\forall (x,y) \in \mathcal{X} \times \mathcal{Y}) \qquad \text{and} \qquad \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} Q(X,Y) = 1. \tag{1.1}$$

The set Q is called a *statistical model*, *probability distribution set*, or a *parametric model* if:

$$\mathcal{Q} = \{Q(X,Y;\theta) | \theta \in \Theta \subseteq \mathcal{R}^k\} \tag{1.2}$$

where $k \in \mathcal{Z}$ is the integer parameter dimension.

It is assumed that for all distributions of interest, for all $(X,Y) \in \mathcal{X} \times \mathcal{Y}$, the parametrization $\theta \rightarrow Q(X,Y;\theta)$ $(\Theta \rightarrow \mathcal{R})$ is *one-to-one* and *infinitely many times differentiable* (sometimes represented by C^∞). This condition is to confirm the existence of derivatives with respect to the parameters θ. Throughout, the notations $\partial_i Q(X,Y;\theta) = \frac{\partial Q}{\partial \theta_i}$ and $\partial_i \partial_j Q(X,Y;\theta) = \frac{\partial^2 Q}{\partial \theta_i \partial \theta_j}$ are used frequently.

1.1.1.1 Fisher information

The Fisher information (FI) plays an important role in estimation theory as a measure of information existing in a set of random variables about the unknown parameter.

In estimation, the inverse of the FI determines an upper–bound on the accuracy attainable in estimation. Additionally, the invariance of this information with respect to re–parametrization in the space of probability distributions is the fundamental concept in the differential geometry approach to statistics, also referred to as *information geometry.* In the following, some useful concepts from information geometry are studied. Beyond these points, the subject of information geometry is not directly relevant for the purpose of this thesis. For more detail on information geometry many excellent references exist, for example [9] and [58].

For the PD $Q(X, Y; \theta)$, the Fisher information matrix (FIM):

$$J(\theta) = [J_{ij}] \quad \forall i, j \in \{1, ..., k\} \tag{1.3}$$

is defined as follows:

$$J_{ij} = \mathrm{E}_\theta[\partial_i \partial_j l(X; \theta)], \tag{1.4}$$

where $l(X, Y; \theta) = \log Q(X, Y; \theta)$ and E_θ is the *expectation* operator with respect to the PD $Q(X, Y; \theta)$.

1.1.1.2 Additivity of Fisher information

Suppose $(x^n, y^n) = (x_1, y_1), ..., (x_n, y_n)$ is a set of $i.i.d$ random samples from a probability distribution $Q(X, Y; \theta)$. It is easy to show that [24]:

$$J_n(\theta) = nJ(\theta), \tag{1.5}$$

where $J_n(\theta)$ is defined as the Fisher information in (x^n, y^n). This property shows that it is sufficient to study the Fisher information of a single observation (the Fisher information is an intrinsic attribute of any probability distribution and therefore is invariant under sampling method, observation method, etc.).

1.1.1.3 Sufficient Statistics

A *statistic* S = h(X, Y) is any function, $h : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{R}$. The statistic S(X, Y) is *sufficient* for parameter θ if $\theta \rightarrow$ S(X, Y) $\rightarrow$ (X, Y) (Markov-chain property), i.e. if $p(X, Y|\mathrm{S(X, Y)})$ is independent from θ. In general the statistic can be a vector-valued function. In this case, the sufficient statistics S(X, Y) = h(X, Y) are sometimes called *joint–sufficient* ([34], page 364). Note that a statistic is in fact an induced random variable S = h(X, Y).

Theorem 1.1.1 *(Factorization Theorem) Given* (x^n, y^n) *as* n *samples drawn from* $Q(X, Y; \theta)$, S *(generally vector valued) is a* sufficient statistic *for* θ *if and only if the* likelihood *of* (x^n, y^n) *factorizes into the following form:*

$$\mathcal{L}(\theta) = q(\theta, S(x^n, y^n)).r(x^n, y^n), \tag{1.6}$$

for some functions q *and* r.

Proof 1.1.1 *Only the proof for the discrete case is presented. Assume that the likelihood factorizes as above. Let define* $Z^n \triangleq (x^n, y^n)$. *For the conditional distribution of* Z^n *given the statistic* S *we have:*

$$\begin{aligned} Q(Z^n|\mathrm{S(Z^n)}) &= \frac{Q(Z^n, \mathrm{S(Z^n)})}{p(S(Z^n))} \\ &= \frac{Q(Z^n)}{\sum_{W^n:\mathrm{S(W^n)=S(Z^n)}} Q(W^n)} \\ &= \frac{q(\theta, \mathrm{S(Z^n)}).\mathrm{r(Z^n)}}{\sum_{W^n:\mathrm{S(W^n)=S(Z^n)}} q(\theta, \mathrm{S(Z^n)}).\mathrm{r(W^n)}} \\ &= \frac{r(Z^n)}{\sum_{W^n:\mathrm{S(W^n)=S(Z^n)}} r(W^n)} \end{aligned}$$

which is not a function of θ. *Here* W *is an arbitrary auxiliary random variable. Conversely assume that* S *is a sufficient statistic for* θ. *Then the likelihood factorizes*

as follows:

$$\begin{aligned}\mathcal{L}(\theta) &= Q(Z^n|\theta) \\ &= Q(Z^n|\mathrm{S}(\mathrm{Z}^\mathrm{n}),\theta)\mathrm{Q}(\mathrm{S}(\mathrm{Z}^\mathrm{n})|\theta) \\ &= r(Z^n).q(\mathrm{S}(\mathrm{Z}^\mathrm{n}),\theta).\end{aligned}$$

Example (Bernoulli Distribution): A sufficient statistics for *Bernoulli*(θ) is $\sum X_i$ since:

$$\mathcal{L}(x^n|\theta) = \prod_i (\theta)^{x_i}(1-\theta)^{1-x_i} = (\theta)^{\sum x_i}(1-\theta)^{n-\sum x_i} = g(\theta, \sum x_i).1.$$

Example (Exponential Distribution): Let x^n be n samples from an exponential probability distribution:

$$f(X|\theta) = r(X)\exp\{\eta(\theta)S(X) - B(\theta)\}.$$

Then $S(x^n) = \sum_i S(x_i)$ is a sufficient statistic for θ.

1.1.1.4 Maximum likelihood (ML) parameter estimation

Suppose $(x^n, y^n) = (x_1, y_1), ..., (x_n, y_n)$ is a set of random samples from a probability distribution $Q(X, Y; \theta)$ with $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ and $\theta \in \Theta$. It is desired to estimate the underlying distribution, i.e. to estimate the parameter θ. As estimator $\hat{\theta}$ is a function from $\mathcal{X}^n \times \mathcal{Y}^n$ to Θ. In ML estimation, the parameter of interest is estimated by maximizing the (log)likelihood of the observed samples:

$$\hat{\theta}_{ML} = \arg\max_{\theta\in\Theta} l(x^n, y^n; \theta),$$

where $l(x^n, y^n; \theta) = \log Q(x^n, y^n; \theta)$ is the log-likelihood of (x^n, y^n). Note that for any given (x^n, y^n), $l(x^n, y^n; \theta)$ is considered as a function of θ. We call $\hat{\theta}_{ML}$ an ML estimator. It is easy to verify that the ML estimator is a function of a sufficient statistic, and therefore it is a *statistic*. Although not generally true, in most problems the ML

estimator is also a sufficient statistic. An important property of ML estimation is the fact that if the ML estimator is sufficient statistic, then it is a *minimal sufficient statistic* [14].

Remark 1 *When complete set of data (x^n, y^n) is available, the solution of ML estimation involves a maximization in (1.1.1.4). On the other hand, when the data is partially available, i.e. the data is* incomplete, *(e.g. when only y^n is available) the ML estimation involves the maximization of a lower bound on the log–likelihood. This is the basis of the expectation–maximization (EM) algorithm explained in the following after reviewing more definitions.*

1.1.1.5 Estimator unbiasedness and efficiency

An estimator $\hat{\theta}_n$ is called *unbiased* when:

$$\mathrm{E}_\theta \hat{\theta}_n = \theta,$$

where E_θ is the expectation operator with respect to $Q(X;\theta)$. In addition, the performance of any estimator is measured by its mean–square error (MSE) defined as:

$$MSE(\hat{\theta}_n) = \mathrm{E}_\theta[(\hat{\theta}_n - \theta)(\hat{\theta}_n - \theta)^T].$$

When $\hat{\theta}_n$ is unbiased the MSE equals the variance–covariance matrix of the estimator $\hat{\theta}_n$ defined as:

$$C_n(\hat{\theta}_n) = \mathrm{E}_\theta[(\hat{\theta}_n - \mathrm{E}_\theta(\hat{\theta}))(\hat{\theta}_n - \mathrm{E}_\theta(\hat{\theta}))^T].$$

The estimator variance $V_n(\hat{\theta}_n)$ is defined respectively. Note that the estimator $\hat{\theta}$ and its variance-covariance matrix are in general functions of the sample size n. The mean square error of an unbiased estimator is lower bounded by the inverse of the Fisher information, also known as the Cramer-Rao lower bound (CRLB).

Theorem 1.1.2 *(Cramer-Rao inequality) The variance–covariance matrix $C_n(\hat{\theta}_n)$ of an unbiased estimator $\hat{\theta}_n$ satisfies:*

$$C_n(\hat{\theta}_n) \geq \frac{1}{n} J(\theta)^{-1},$$

in the sense that $nC_n(\hat{\theta}_n) - J(\theta)^{-1}$ is positive semidefinite.

Proof 1.1.2 *Refer to ([9], page 42).*

For the scalar parameter case, the inequality for the variance of estimation is as follows:

Theorem 1.1.3 *(Cramer-Rao inequality- Scalar parameter) The variance $V_n(\hat{\theta}_n)$ of an unbiased estimator $\hat{\theta}_n$ is lower bounded by the inverse of the Fisher information:*

$$V_n(\hat{\theta}_n) \geq \frac{1}{nJ(\theta)},$$

where:

$$J(\theta) = \mathrm{E}_\theta\big[\frac{\partial}{\partial\theta} l(X;\theta)\big]^2$$

Proof 1.1.3 *Refer to ([24], page 328).*

The estimator is called *efficient* if its variance achieves the minimum achievable variance.

1.1.1.6 Estimator asymptotic consistency and efficiency

Here, the asymptotic behavior of an estimator is studied in the limit $n \to \infty$. In this case, an estimator, or more precisely a sequence of estimators $\{\hat{\theta}_n, n = 1, 2, ...\}$ is called *asymptotically consistent* if for any θ, the estimate $\{\hat{\theta}_n\}$ converges in probability to θ as $n \to \infty$. In other words, for any arbitrarily small $\epsilon > 0$:

$$\lim_{n\to\infty} \Pr\Big\{|\hat{\theta}_n - \theta| > \epsilon\Big\} = 0.$$

As n becomes sufficiently large, the probability distribution of such an estimator is concentrated around θ and therefore for all values of the parameter θ the expectation of the estimator converges to θ uniformly:

$$\lim_{n\to\infty} \mathrm{E}_\theta \hat{\theta} = \theta.$$

The mean square error of an asymptotically consistent estimator is lower-bounded by the CRLB as in the case of a consistent estimator:

$$\lim_{n\to\infty} nC_n(\hat{\theta}_n) \geq J(\theta)^{-1}.$$

A consistent estimator that achieves (as n becomes sufficiently large) the equality in CRLB inequality for all θ is called *asymptotically efficient.*

1.1.1.7 Consistency and efficiency of ML estimation

Although the minimum achievable variance is not defined for general estimators, it is defined for the class of ML estimators. The importance of the ML estimation procedure becomes apparent in the following theorem.

Theorem 1.1.4 *(Efficiency of ML estimation) The maximum likelihood estimator $\hat{\theta}_{ML}$ is asymptotically consistent and efficient:*

$$\lim_{n\to\infty} nC_n(\hat{\theta}_{ML}) = J^{-1}(\theta).$$

More precisely, the probability distribution of $\hat{\theta}_{ML}$ is a Gaussian with mean θ and covariance $\frac{1}{n}J^{-1}(\theta)$.

Proof 1.1.4 *Refer to ([9], page 84).*

The result of this theorem is the main reason why we are interested in ML estimation in a distributed scenario in the following chapters.

1.1.1.8 Loss in Fisher information

Due to the important role that the FI plays in ML estimation, it is necessary to learn whether a sufficient statistic conveys the FI completely. Also, it is useful to study the FI loss when the sufficient statistic is not completely available. The following theorem [9] provides the answer to such questions.

Theorem 1.1.5 *(Efficiency and Fisher information) Suppose $S = h(X)$ is a statistic (possibly vector valued) of θ. Also suppose that the PD $Q(X;\theta)$ is trivially factorized in the form: $Q(X;\theta) = q(h(X);\theta)r(X;\theta)$ where $q(S;\theta) = q(h(X);\theta)$ is the statistical model of the random variable S induced by the mapping function h. The Fisher information of the induced probability distribution $Q(S;\theta)$, $J_h(\theta)$, is upper-bounded by the Fisher information of the original PD $Q(X;\theta)$, i.e. $J_h(\theta) \leq J(\theta)$, in the sense that $\Delta J(\theta) = J(\theta) - J_h(\theta)$ is* positive semidefinite. *A necessary and sufficient condition for the equality $J_h(\theta) = J(\theta)$ to hold is that h is a sufficient statistic for θ. The information loss $\Delta J(\theta)$ is given by:*

$$\begin{aligned} \Delta J_{ij}(\theta) &= \mathrm{E}_\theta[\partial_i \log r(X;\theta)\partial_j \log r(X;\theta)] \\ &= \mathrm{E}_\theta\Big[Cov[\partial_i l(X;\theta)\partial_j l(X;\theta)|S]\Big], \end{aligned}$$

where $\mathrm{E}_\theta\Big[Cov[\partial_i l(X;\theta)\partial_j l(X;\theta)|s]\Big] = \int Cov[.,.|s]q(s;\theta)ds$ and $Cov[.,.|s]$ for a fixed s denotes the covariance with respect to the conditional distribution $p(X|Y;\theta)$.

Proof 1.1.5 *See Appendix A.*

1.1.2 ML estimation using incomplete data: the EM algorithm:

Suppose that sufficient statistics for estimation of θ is not available completely. The partial data, here y^n, is usually referred to as the *incomplete data*. In this case,

the ML estimation involves the maximization of a lower bound on the log–likelihood function. Let $U(X;\theta)$ denote a variational distribution, generally a function of θ, over the input space. The log–likelihood of the available data can be written:

$$\begin{aligned}
\mathcal{L}(\theta, y^n) &= \log Q(y^n;\theta) \\
&= \log \sum_{x^n} Q(x^n, y^n;\theta) && (1.7) \\
&= \log \sum_{x^n} U(X;\theta)\frac{Q(x^n, y^n;\theta)}{U(X;\theta)} \\
&\geq \sum_{x^n} U(X;\theta)\log\frac{Q(x^n, y^n;\theta)}{U(X;\theta)} && (1.8) \\
&= \sum_{x^n} U(X;\theta)\log Q(x^n, y^n;\theta) - \sum_{x^n} U(X;\theta)\log U(X;\theta) \\
&= \sum_{x^n} U(X;\theta)\log Q(x^n, y^n;\theta) + H(U) && (1.9) \\
&\triangleq \mathcal{F}(\theta, U), && (1.10)
\end{aligned}$$

where in (1.7) the likelihood is summed (integrated for continuous–valued variables) over the unknown x^n. In (1.8) Jensen's Inequality [24] is used. Here $H(U)$ is the entropy of the distribution $U(X;\theta)$.

As can be seen from (1.9), the likelihood of the available data is lower–bounded by the expectation of the likelihood of the complete–data over the unknown variable and the entropy of the variational distribution $U(X)$.

The EM algorithm attempts to maximize this lower bound in an iterative fashion. In the E-step at iteration t, the unknown parameter θ_t is assumed to be known and fixed. The $\mathcal{F}$ function (1.10) is maximized with respect to $U(X;\theta_t)$:

$$\textit{E-step:}\quad U_{t+1} = \max_{U} \mathcal{F}(\theta_t, U_t) \tag{1.11}$$

where $U_t = U(X;\theta_t)$ and U_{t+1} is the optimal argument obtained in the optimization.

In the M–step, $\mathcal{F}_{\theta,U_{t+1}}$ is maximized with respect to θ to obtain a new value for

the parameter:

$$\textit{M-step:} \quad \theta_{(t+1)} = \arg\max_{\theta} \mathcal{F}(\theta, U_{t+1}). \tag{1.12}$$

The EM algorithm alternates between these two steps until convergence is reached.

Different variants of the EM algorithm perform different versions of the E–step. For instance, in the classical version of the EM algorithm [35] the E–step involves computation of the posterior distribution of x^n given the available data y^n:

$$U^*(X;\theta_t) = p(x^n|y^n;\theta_t). \tag{1.13}$$

One can verify that, in this case, the variational distribution $U^*(X;\theta_t)$ is optimal, i.e. achieves the lower-bound in (1.8). In this case, the E–step is in fact the computation of the expectation of the log-likelihood of the complete–data distribution when the parameter is assumed to be known θ_t.

Remark 2 *The EM algorithm will be used in Chapter 3 to implement a distributed estimator using side–information. The EM algorithm also will be studied in Chapter 4 to verify the results of an information geometric algorithm for channel estimation in wireless MIMO channels. In Chapter 5 a special implementation of the algorithm is used to solve a joint estimation and identification problem in non–linear state space models. In that chapter, the E–step is implemented by a particle smoother and the estimation of the variational estimation M–step is replaced with a non–linear regression using a mixture–of–Gaussians.*

In all these implementations, the E–step is involved with computation of the posterior distribution (1.13).

1.2 Communications for estimation

In distributed coding of two correlated discrete sources, the objective is to reconstruct the sources' symbols at a common decoder. In contrast, in distributed estimation, the

main goal of coding is to protect the relevant information about the source parameters of interest.

In a "communications Utopia", optimality is defined by accuracy in estimation and the objective of optimization in a transmission system is the "Fisher information" rather than the bit–error–rate.

1.2.1 Motivation: A distributed weather network

It was Shannon who through his source coding theorem [24] showed that a code with rate R at least equal to the entropy of the source $H(X)$ is necessary and sufficient to noiselessly represent the information content of the source X. Moreover, in a theorem known as the *"separation theorem"* [24] he showed that in a point-to-point system the communications with arbitrarily small error is possible if and only if the channel capacity is at least equal to the rate of the source; $C \geq R$. Therefore, one of the key questions in source coding is how much rate is sufficient and necessary to preserve the source content within the context of communications.

It was believed that in a distributed source coding scenario with two correlated sources with code rates R_x and R_y respectively, a sum rate of $R = R_x + R_y \geq H(X) + H(Y)$ was necessary to encode the two sources noiselessly. However, in their seminal paper, Slepian and Wolf (SW) [90] showed that, surprisingly, by means of using the correlation information between the sources, one can do much better than that and a sum rate of $R = H(X) + H(Y|X) = H(X, Y)$ is sufficient for this purpose.

In the distributed estimation case, rather than the reproduction of the sources, the objective is estimation of the source parameter. The Slepian–Wolf rates are sufficient for such a purpose (the parameter can be estimated using the reconstructed sources). However it is not clear whether one can do better than these rates for parameter estimation. The first question raised in this thesis is whether the SW rates are also *necessary* for distributed parameter estimation. To provide further motivation, we

provide the following example from [24]. We then extend the example.

Example 14.4.1 ([24], page 409) Consider the weather in Gotham and Metropolis represented by binary random variables X and Y respectively. For the purposes of our example, we assume that Gotham is sunny with probability 0.5 and that the weather in Metropolis is the same as in Gotham with probability 0.89. The joint distribution of the weather is given as follows:

$p(X,Y)$	Metropolis Rain	Shine
Gotham		
Rain	0.445	0.055
Shine	0.055	0.445

Assume that we wish to transmit 100 days of weather information to the National Weather Service Headquarters in Washington. We could send all the 100 bits of the weather in both places, making 200 bits in all. If we decided to compress the information independently, then we would still need $100H(0.5) = 100$ bits of information from each place for a total of 200.

If instead we use the coding scheme prescribed by the Slepian-Wolf theorem we need only $nH(X) + nH(Y|X) = 100H(0.5) + 100H(0.89) = 100 + 50 = 150$ bits in total. According to the same theorem, $nH(Y|X) = 100 * H(0.89) = 50$ bits need to be sent from each of the cities to the other one in order for both cities to have all the weather information.

Example cont'd: number of days with different weather We now assume that the only desired information at the headquarters is the number of days during which the weather in both cities are the same (or different). The answer to such a question is the main subject of Chapter 2. In this case, it will be shown that even though we do not need to reconstruct all the information at the headquarters, we still need

at least $nH(X) + nH(Y|X) = 150$ bits to be sent. Notice that this is in contrast to the intuition that "*since only the correlation information is needed to be sent to the headquarters, a sum of only* $100I(X;Y) = 100(H(X) - H(X|Y)) = 50$ *would be sufficient.*"

1.3 Problem Statement

Suppose a phenomenon is characterized by a *discrete* probability distribution $Q(X, Y; \theta)$, where the random variables X and Y take discrete values from sets of discrete alphabets $\mathcal{X}$ and $\mathcal{Y}$, respectively and where $\theta \in \Theta \subseteq \mathcal{R}^k$ is a parameter vector by which the PD is uniquely identified. Let also (x^n, y^n) be n *i.i.d* samples drawn from $Q(X, Y; \theta)$. The sequences are encoded into separate messages by encoding functions $f : \mathcal{X}^n \rightarrow \Pi_f$ and $g : \mathcal{Y}^n \rightarrow \Pi_g$, respectively. The message sets Π_f and Π_g, also referred to as the *codebooks*, are arbitrary discrete sets $\Pi_f = \{1, 2, ..., 2^{nR_x}\}$ and $\Pi_g = \{1, 2, ..., 2^{nR_y}\}$. Here R_x and R_y are called the *code rates* defined as $R_x = \lim_{n\rightarrow\infty} \frac{\log|\Pi_f|}{n}$ and $R_y = \lim_{n\rightarrow\infty} \frac{\log|\Pi_g|}{n}$, where $|\Pi_f|$ and $|\Pi_g|$ are the cardinality of the sets Π_f and Π_g, respectively.

The encoded messages are transmitted to a common receiver via separate communication channels. The communications channel capacities are generally limited. Thus the design of the codebooks and the messages involve compression. By appropriately incorporating the compression such that the rate of the codes is "no more" than the available capacities of the channels, the decoder receives the noiseless encoded messages. At the receiver, by means of a decoder/estimator function $h : \Pi_f \times \Pi_g \rightarrow \Theta$, it is desired to estimate the parameter of the underlying distribution, $\hat{\theta} = h(f(x^n), g(y^n))$ (see Figure 1.1).

The pair of rates (R_x, R_y) is called *achievable* if there exists at least one pair of encoders (f, g) and a decoder h with probability converging to 1 by which one can

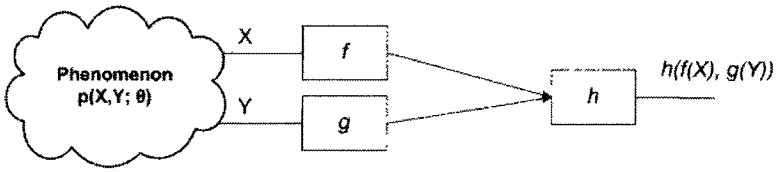


Figure 1.1: Distributed estimation

construct the sequences of codes that provide transmission of *sufficient statistics* for the parameter θ to the receiver with probability of error converging to 0 as n becomes sufficiently large.

It is desired to determine the region of achievable rates for distributed estimation. It is also useful to determine the rates under which the distributed estimator *can achieve* the same accuracy as it can in *local estimation.* Moreover, it is desired to design communication systems that are able to achieve the local accuracy bounds.

Remark 3 *Throughout this thesis, we refer to estimation as being* local *(or centralized) when it is performed using completely available sufficient statistics. This is in contrast to distributed estimation where in general sufficient statistics undergo compression and encoding. Also, an estimator is called* efficient *when its variance achieves its lower bound (e.g. Cramer-Rao lower bound).*

1.3.1 A Brief Review of Literature

Multiterminal estimation was first introduced by Toby Berger in [15] and then elaborated on by Zhang and Berger (ZB) in [106]. In these interesting papers, given a set of positive rates, it was demonstrated that there always exists asymptotically unbiased estimators for distributed parameter estimation. A single–letter upper bound on the maximum accuracy that the optimal estimator can achieve was also established. The main weakness of the ZB approach was a limiting constraint on the joint distributions

of correlated variables, referred to as the "additivity condition", which does not hold in general nor under parameter transformations.

Later, Amari [6] solved the distributed estimation problem under *zero-rate* compression, from an information-geometric point of view. In this contribution, he showed that when the marginal distribution(s) are function(s) of the unknown parameters, asymptotically efficient estimation is possible. He also developed the maximum likelihood (ML) estimator based on the observed marginal *types* (defined in later chapters) and computed the achievable accuracy, enumerated by the *observed Fisher information.* His method, however, does not include positive rates.

In the first attempt to solve the positive–rate distributed estimation, Ahlswede and Burnashev [1] chose the *min-max* approach to estimation of a scalar parameter using full side–information (SI), provided that the marginal distribution of the SI was independent of the parameter (i.e. an ancillary statistic). They showed that there exist efficient estimators whose *min-max* variance index can achieve the *max-min* Fisher information. They also computed both the variance index (as a function of the available rate) and the Fisher information.

Perhaps the most important contribution in the field is due to Han and Amari [47] who solved the distributed estimation problem for arbitrary positive rates and vector–valued parameters. The common method used by these authors is based on the introduction of auxiliary random variables that form a Markov chain with the original random variables. An asymptotically unbiased effective estimator using only the marginal joint–types of these auxiliary random variables is constructed. Then the constructed ML equations are solved and solutions based on the transmitted joint–type are obtained. Also, the achievable Fisher information as a function of the available rates and the unknown parameters is computed using the ML estimation.

Remark 4 *The method of Han and Amari (HA) is the most recent result on distributed estimation (mainly on two-terminal estimation). It is shown by the authors*

that this method not only relaxes the additivity condition constraint in the ZB method, but provides a substantially smaller variance index. Also, it is shown that these results can be simplified to the min-max *results of Ahlswede and Burnshev [1]. Therefore, in the following chapters, when a comparison to the literature is necessary, we refer solely to the method of Han and Amari [47] [48].*

Remark 5 *One of critical steps in the HA method is the appropriate choice of test channel distributions (explained later in Section 2.8.1) that depends on the particular problem in hand. The choice of these test channels also has a prominent effect of the region of achievable rates, as well as the accuracy that can be attained by the ML estimator.*

More importantly, when such selections are made, the calculation of certain important parameters in the HA method, e.g. the matrix $H_{M'}$ of the projection on the observable types (c.f. Section 2.8.1), is non-trivial. This is the case even for the simplest case of binary symmetric sources ([47]). These complications make this solution mathematically intractable, and hence inaccessible to the engineering community (as noted by the same authors in [48], and also noted in [55]) .

1.3.2 Thesis contribution:

We approach the problem from a different perspective by answering the question: "How much rate (or sum of rates) is necessary to have efficient estimation?". We show in Chapter 2 that for fixed parameters, the rates in the Slepian–Wolf region are not only sufficient but also *necessary* for *efficient* estimation. In order to prove the main theorem, we also devised a novel distributed binning scheme and a new theorem in large deviation theory that are used for proving our distributed coding theorem. The proof for the converse is implemented using a generalization of Fano's inequality for distributed scenarios [114] [110].

As we show in the last part of that chapter, determination of the region of achievable rates for efficient estimation of a general source is an extremely difficult problem. This fact is the motivation for proposing methods that provide practical guidelines for designing distributed estimation systems. One example of such approaches proposed in [88] for binary symmetric sources. Given any source parameter for a binary symmetric source, this theorem determines the region of achievable rates for efficient estimation of the source.

We generalize this theorem for a larger class of sources. More specifically, we provide a lower bound on the region of achievable rates (i.e., existence of encoder/decoders for attaining an accuracy equivalent to local estimation) for estimation of sources with a convex mutual information with respect to the unknown parameter θ.

Finally, we study a famous *"Modulo-two adder" source network* due to [61]. We show that in such a network, if communications is for the purpose of estimation, in contrast to the Slepian–Wolf rates, the triple set of $(0, 0, 0)$ rates are achievable.

Our approach is closely related to the distributed source coding theorem first proved by Slepian-Wolf [90]. In fact, we extend the distributed source coding theorem for the special case in which the accuracy in parameter estimation (noiseless transmission of joint–type) is the main concern, rather than the perfect reconstruction of the sources per se.

In Chapter 3 an efficient implementation of universal coding schemes for distributed estimation of a binary symmetric source is presented [115] [111]. Since the correlation channel between the sources is assumed to be unknown at the joint decoder, previously proposed distributed coding schemes are not useful for this purpose. Therefore, LDPC-based coset–coding schemes are extended to the case where the correlation channel is unknown at the decoder. The basic idea is to implement the EM algorithm on a factor–graph that includes an LDPC decoding mechanism.

1.3.3 Relationship with Distributed Coding

Distributed source coding involves the study of the theoretical limits and practical techniques for encoding (and quantization) and optimal reconstruction (with respect to a fidelity criterion) of two or more correlated sources, when the communication resources (rates) are limited. The main challenge that discriminates distributed source coding from point-to-point source coding is that the encoders have no (or limited) collaboration.

From the information theoretical point of view, the distributed estimation literature is divided into two different but closely related problems; i.e. distributed source coding (source estimation) and distributed parameter estimation (multiterminal estimation). The key difference between the two categories corresponds to the definition of estimation; estimation (reconstruction) of the source(s) in the former and parameter estimation (reconstruction of function(s) of the source(s)) in the later case. The current main stream of research seems focused on the distributed source coding. There are a large number of references for distributed coding, of which ([104], and the references therein) is more related to our interests.

1.3.4 Relation to network information theory

Shannon's *separation principle* [24] separates the necessary and sufficient conditions of noise-free *"point-to-point"* communications for transmission of source symbols with a given distortion. This principle lets communications engineers work in two related but separate communities, i.e. the source-coding and channel coding communities. However, there are particular circumstances, like a network of sources and receivers where the separation principle is not valid anymore. A counterexample [23] shows that the *source-channel separation theorem* does not hold in general for networks of sources. In that example, it is shown that although the rates of source information are

more than the channel capacity, error-free transmission is possible (in contradiction to the *"necessary condition"* part of the *separation principle*).

Throughout, we assume that the source code rates are appropriately chosen (less than the capacities of the channels), and therefore once the sources are encoded, they can be transmitted losslessly. This can be guaranteed, for example, by assuming that the transmission is via a set of orthogonal multiple-access channels for which case the sufficiency and necessity of the separation theorem is proved [103].

1.4 Information geometric approach to MLE

1.4.1 Motivation

In Chapter 4, semi–blind channel estimation in MIMO wireless communications is posed as an incomplete–data problem. The EM algorithm has been used in different varieties for this purpose. The current algorithms in the literature suffer a very slow convergence rate with complexity of computations exponentially growing with the length of channel (in ISI channels) or the size of constellations.

The main goal in Chapter 4 is to propose a fast semi–blind channel estimation algorithm relative to previously proposed algorithms. Here bold lower-case symbols indicate a vector quantity while a symbol in calligraphic style indicates a set of probability distributions. The subscript t is the iteration index and k is the temporal index.

1.4.2 Problem statement

We consider the following linear time-invariant MIMO system with M transmitters and N receivers:

$$\boldsymbol{y}(k) = \sqrt{\frac{\rho}{M}}\boldsymbol{H}\boldsymbol{x}(k) + \boldsymbol{v}(k)$$

where $\boldsymbol{y}(k) \in \mathbb{C}^N$ and $\boldsymbol{x}(k) \in \Omega^M$ are the output and the input vectors, respectively, k is the time index, and Ω is a complex constellation with C members, such that the average energy over all members of the constellation is unity. The quantity $\boldsymbol{H} \in \mathbb{C}^{N \times M}$ is the complex channel coefficient matrix, whose elements are zero mean random variables, scaled to unit *rms* values. The quantity ρ is the SNR on each receive channel. Also, the sources are chosen to be *i.i.d.*, whose components are zero–mean Gaussian. The quantity $\boldsymbol{v} \sim N(\boldsymbol{0}, \boldsymbol{\Psi})$ is the noise vector with generally unknown covariance $\boldsymbol{\Psi} \in \mathbb{C}^{N \times N}$. It is assumed that $\boldsymbol{\Psi}$ is full rank.

The above MIMO model is valid in an intersymbol–interference (*ISI*) free Rayleigh–fading channel. It is assumed the channel $\boldsymbol{H}$ and the covariance $\boldsymbol{\Psi}$ are constant over a block of L transmitted symbols. This model is useful in space–time coding systems, where in many cases it is necessary to (semi) blindly identify the channel [4, 93, 94]. This model is also widely adopted in OFDM systems, e.g., [5].

A joint *pdf* of the input and output variables, e.g. $q(\boldsymbol{z}; \boldsymbol{\theta})$, where $\boldsymbol{z} = [\boldsymbol{y}^T, \boldsymbol{x}^T]^T$ is the *complete* data, and $\boldsymbol{\theta} = (\boldsymbol{H}, \boldsymbol{\Psi})$ is the parameter set, provides a complete description of the underlying signal model. In general, for a given $\boldsymbol{z}$ there exists a one–to–one correspondence between $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ and $q(\cdot\,; \boldsymbol{\theta}) \in \mathcal{Q}$, where $\boldsymbol{\Theta}$ is the parameter space, and $\mathcal{Q}$ is the set of likelihood distributions, defined by

$$\mathcal{Q} = \{q(\boldsymbol{z}; \boldsymbol{\theta}), \boldsymbol{\theta} \in \boldsymbol{\Theta}\}.$$

The ML estimation task is then to choose a distribution in this family that best describes the complete data. By assuming that L independent complete data samples $\boldsymbol{z}_k$, $k = 1, \ldots, L$ are available, the maximum likelihood estimation problem is to find the distribution $q^*(\boldsymbol{z}; \boldsymbol{\theta}^*)$ that satisfies

$$q^*(\boldsymbol{z}; \boldsymbol{\theta}^*) = \arg \max_{q \in \mathcal{Q}} \prod_{k=1}^{L} q(\boldsymbol{z}_k; \boldsymbol{\theta}).$$

There are situations where the complete data are only partially available, i.e., we observe only $\boldsymbol{y}$. In these circumstances, the question is how to maximize the

likelihood of observations and select the distribution $q(\boldsymbol{z};\boldsymbol{\theta}) \in \mathcal{Q}$ given only the partially available data (also called *incomplete data*). Assuming that the input is discrete and distributed according to the *pdf* $p(\boldsymbol{x})$, one must solve the following equivalent *incomplete-data* problem:

$$q^*(\boldsymbol{y};\boldsymbol{\theta}^*) = \arg\max_{q\in\mathcal{Q}} \prod_{k=1}^{L} \sum_{\boldsymbol{x}_k} q(\boldsymbol{y}_k|\boldsymbol{x}_k;\boldsymbol{\theta}) p(\boldsymbol{x}_k).$$

1.4.3 The EM-based channel estimation algorithms

Early work on blind channel identification using the stochastic ML estimation (specifically EM algorithm) was by Feder [40]. In [57] the EM algorithm is used for jointly estimating the channel and detecting the symbols in a single-input single-output (SISO) system in an ISI dispersive channel. For a geometric derivation of the EM algorithm as well as generalized successive interference cancelation algorithms for CDMA channel estimation refer to [54]. The EM algorithm was also used for identification of the channel in MIMO systems with OFDM modulation in [68]. Similar results for multi-input single-output (MISO) systems were given in [5]. The intensive computations necessary for the E-step makes the algorithms very slow in convergence. Therefore, these algorithms are usually computationally very slow. Besides, their computational complexity increases with either the length of channel (in ISI channels) or the number of constellation points.

Several attempts have been made to speed up the convergence. Examples are the space-alternating generalized EM (SAGE) in which the algorithm alternates between several hidden spaces rather than using one "complete" data space, and therefore, instead of all the unknown parameters, a subset of them are being updated in each iteration [41]. For a comprehensive application of this algorithm for joint detection and channel estimation in a multiuser DS-CDMA system refer to [60]. The EM and SAGE algorithms are also used for channel estimation in a space-time coded OFDM

with transmit diversity [100].

The EM-based algorithm for semi–blind channel estimation implemented in this thesis (summarized in Appendix E) is included for the sole purpose of comparison of the results from a proposed algorithm based on information geometry. The algorithm used is the essential algorithm used in many other references for similar purposes. For some examples refer to [5], [40], [60], [105], [54], [26], [71], [41], [100], and [68].

1.4.4 Thesis contribution

In Chapter 4 we pose the incomplete data problem in an information geometric framework [28]. Information geometry encompasses a theoretical framework for a better understanding of estimation problems. Based on information geometry, a low-complexity iterative identification procedure, called the *IGID algorithm*, for blind identification of unknown parameters in a multi-input multi-output (MIMO) system with Gaussian distributed noise was proposed [112] [113] [109]. The algorithm is an iterative solution to the *incomplete-data problem* posed by maximum likelihood (ML) estimation of parameters in a linear Gaussian MIMO system when only the output observations are available. The IGID algorithm involves two iterative minimizations, corresponding to projections onto the likelihood PD (*probability distribution*) set and the empirical PD set, respectively. A Gaussian assumption on the source allows us to develop closed–form expressions for the projection operations. The performance of the IGID algorithm in blind identification of the channel gain matrix in a MIMO communication system is investigated. It is shown by simulation that the performance of the IGID algorithm is only slightly degraded relative to that of previous EM-based algorithms [5]; however, a noticeable improvement in computational cost is realized.

1.5 Joint identification and estimation using Bayesian filters and MCMC

1.5.1 Motivation

In most solutions to state estimation problems, e.g., target tracking, it is generally assumed that the state transition and measurement models are known a priori. However, there are situations where the model parameters or the model structure itself are not known a priori or are known only partially. In these scenarios, standard estimation algorithms like the Kalman filter and the extended Kalman Filter (EKF), which assume perfect knowledge of the model parameters, are not accurate.

The application of Bayesian (particle) filters as well as the related sampling methods (e.g. Markov-chain Monte-Carlo methods) for joint estimation and identification in nonlinear in the presence of uncertain (and probably non–linear) models is presented in Chapter 5.

1.5.2 Problem statement

State estimation in a nonlinear state-space dynamical system whose evolution process is described as

$$\boldsymbol{x}(t+1) = \boldsymbol{f}(\boldsymbol{x}(t)) + \boldsymbol{u}(t),$$

consists of estimating the state data vector $\boldsymbol{x}$ using a sequence of noisy measurements given by the following model:

$$\boldsymbol{z}(t) = \boldsymbol{h}\big(\boldsymbol{x}(t), \boldsymbol{\theta}\big) + \boldsymbol{v}(t), \qquad t = 1, 2, \ldots,$$

where t is the *discrete* time index, $\boldsymbol{x}(t) \in \mathbb{C}^M$ and $\boldsymbol{z}(t) \in \mathbb{C}^J$ are the state variable and the noisy output measurement vectors respectively, and $\boldsymbol{u}(t) \in \mathbb{C}^M$ is assumed to be an *i.i.d* noise processes, whose probability density function is assumed known and

possibly non–Gaussian. The vector $\boldsymbol{v}(t) \in \mathbb{C}^J$ is a zero-mean Gaussian noise variable with unknown covariance $\boldsymbol{Q}$. The noise $\boldsymbol{v}(t)$ is assumed uncorrelated in time; i.e., $E\big(\boldsymbol{v}(t_1)\boldsymbol{v}(t_2)\big) = \delta_{t_1,t_2}\boldsymbol{Q}$.

Also, the vector valued functions $\boldsymbol{f} : \mathbb{C}^M \mapsto \mathbb{C}^M$, and $\boldsymbol{h} : \mathbb{C}^M \mapsto \mathbb{C}^J$ are assumed to be smooth but otherwise are arbitrary. We assume that the function $\boldsymbol{f}(\cdot)$ is known, whereas uncertainty may exist in the observation model $\boldsymbol{h}(\cdot)$.

A major focus of Chapter 5 is on how to model the partially known or unknown function $\boldsymbol{h}(\cdot)$. If a model which takes into account any known structure in the measurement process is available, then that model should be used in the proposed method, as described later. Any uncertainty is expressed in a parameter vector $\boldsymbol{\theta}$. On the other hand, it is also possible to assume no structure on $\boldsymbol{h}(\cdot)$, as is done with our examples in Sections 5.4 and 5.5. We model this function as an mixture of Gaussians (MoG), again parameterized by the vector $\boldsymbol{\theta}$, in a manner to be described later in Sect. 5.3.2.

1.6 Contribution of the thesis

In Chapter 5 in order to estimate the states in the presence of model uncertainty, we use the *variational* form of the EM algorithm. [108] [107]. The expectation (E) step is implemented by a particle filter that is initialized by a Monte-Carlo Markov chain algorithm. Within this step, the posterior distribution of the states given the measurements, as well as the state vector itself, are estimated. Consequently, in the maximization (M) step, we approximate the nonlinear observation equation as a MoG model. During the M-step, the MoG model is fit to the observed data by estimating a set of MoG parameters. The proposed procedure, called EM-PF (*expectation–maximization particle filter*) algorithm, is used to solve a highly nonlinear bearing-only tracking problem, where the model structure is assumed unknown *a priori*. It is shown that the algorithm is capable of modeling the observations and accurately

tracking the state vector. Additionally, the algorithm is also applied to the sensor registration problem in a *multi–sensor fusion* scenario. It is again shown that the algorithm is successful in accommodating an unknown nonlinear model for a target tracking scenario.

Chapter 2

On Multiterminal Estimation Rates

2.1 Introduction

Suppose two *non–collaborating* correlated sources transmit a pair of sequences via two separate communication channels to a common receiver. The communication channels, in general, have limited transmission capacity. Thus, coding of the sequences involves compression. In the sequel, we assume that the source code rates are appropriately chosen (smaller that the capacities of the channels), and therefore once the sources are encoded, they can be transmitted losslessly.

In distributed *lossless* coding, the objective is to reconstruct the source symbols at the common decoder. The region of achievable rates for this case was studied by Slepian and Wolf (SW) [90]. In contrast, in distributed estimation the main goal of communications is to protect the relevant information about the parameters of interest. The notion of achievability, in this case, is different than what it is in distributed coding. More specifically, it is desired to determine the rates under which the estimator *can achieve* the same accuracy in *distributed* estimation as it can in

local estimation, i.e. when sufficient statistics are completely available at the decoder.

Determination of the region of rates in distributed estimation (also known as multiterminal estimation) is a non–trivial practice, since the region is in general a function of the unknown parameters. The most recent results in distributed estimation due to Han and Amari (HA) [48] provide *sufficient* conditions for achieving efficient distributed estimation. For *fixed parameters*, the SW rates also are *sufficient* for efficient estimation (i.e. the parameters can be estimated using the reconstructed sequences). However, it is not obvious whether these rates are also *necessary*. In other words, for fixed parameters, it is not clear whether one can do better than the SW compression if it is desired to estimate a parameter rather than to reconstruct the sources.

The result presented in this chapter is an effort to answer the question: "what (sum of) rates is *necessary* to have efficient distributed estimation?". Although the question itself is limited to the lossless case, it provides practical insight into the general problem. The approach presented here is an extension of the Slepian–Wolf distributed source coding theorem to the special case where the accuracy in parameter estimation is the main concern. In this chapter we show that for fixed parameters, the rates in the Slepian–Wolf region are not only sufficient but also necessary for efficient estimation. The simple proof based on the method of types and large deviation theory leads to derivation of a lower–bound on the region of rates for a large class of bivariate sources.

2.1.1 A Brief Review of Literature

Distributed estimation was first introduced by Berger [15] and later elaborated on by Zhang and Berger (ZB) [106]. In their interesting paper, they demonstrated that given a set of positive rates, there always exists asymptotically unbiased estimators. They also established a single-letter upper bound on the maximum accuracy that the optimal estimator can achieve. The main weakness of the ZB approach was a limiting

constraint on the joint distributions of correlated variables, referred to as "additivity condition", which does not hold in general nor under parameter transformations.

Later, Amari [6] solved the distributed estimation problem under *zero-rate* compression from an information-geometric point of view. He showed that when the marginal distribution(s) are function(s) of the parameter, asymptotically efficient estimation is possible. He also proposed maximum likelihood estimation of parameters based on the observed marginal types and computed the achievable accuracy, enumerated by the *observed Fisher information.* His method, however, does not include positive rates.

In the first attempt to solve the positive–rate distributed estimation, Ahlswede and Burnashev (AB) [1] chose the *min-max* approach to the estimation of a scalar parameter using full side–information, for the case where marginal distribution of the side–information is independent of the parameter (i.e. it is an ancillary statistic). They showed that there exist efficient estimators whose *min-max* variance can achieve the *max-min* Fisher information. They also computed both the variance (as a function of the available rate) and the Fisher information.

Perhaps the most important contribution in the field is due to Han and Amari (HA) [47] who solved the distributed estimation problem for arbitrary positive rates and vector–valued parameters. For a two-terminal distributed estimation scenario, they demonstrated that provided a set of rate–compatibility conditions are satisfied, there always exists a set of encoders and a decoder to perform asymptotically efficient estimation. Using the marginal types of a set of auxiliary random variables, they also constructed a maximum likelihood (ML) estimation equation. The constructed equation was used to solve for the ML estimation of the unknown parameter. Moreover, it provided the attainable accuracy (i.e. Fisher information), as a function of the available rates as well as the unknown parameter. The definition of the rate–compatibility

conditions was to guarantee that sufficient information about the joint–type of correlated random variables is transmitted to the decoder (to be able to estimate the parameter).

Remark 1 *The method of Han and Amari (HA) [47] [48] is the most recent result on distributed estimation (mainly on two-terminal estimation). It is shown by the authors that this method not only relaxes the additivity condition constrained by Zhang and Berger [106], but provides a substantially smaller variance. Also, it is shown that these results simplify to the* min-max *results of Ahlswede and Burnsev [1]. Therefore, when a comparison to literature is necessary, we refer solely to the HA method.*

2.1.2 Thesis contribution:

The rates in the SW region are sufficient for reconstruction of correlated sequences in distributed coding. Therefore, these rates are also sufficient for efficient parameter estimation, i.e. the parameters can be estimated using the reconstructed sequences. In addition, the HA method provides sufficient rates for this purpose. However, it is not obvious whether these rates are also *necessary*. In other words, these methods do not answer the question whether one can do better than the Slepian-Wolf compression if it is desired to estimate a parameter rather than to reconstruct the sources.

For instance, as the sole complete solution to the problem, one of the critical steps in the HA method is the appropriate selection of the *test channel distributions* (conditional distributions that relate the correlated random variables to their corresponding auxiliary random variables, explained in detail later in Section 2.8.1). The selection process differs for any particular problem under consideration and has a prominent effect on the region of achievable rates as well as the accuracy that can be attained by the ML estimator. Moreover, there is no systematic method for this process. More

importantly, when such selections are made, the calculation of the important parameters in the HA method, e.g. the matrix $H_{M'}$ of the projection on observable types (c.f. Section (2.8.1)) is non-trivial, even for the simplest case of binary symmetric sources ([47]). These complications make the HA solution mathematically intractable and hence inaccessible to the engineering community (as noted by the same authors in [48]), also in [55].

We approach the problem from a different perspective by answering the question: "how much rate (or sum of rates) is necessary (and sufficient) to have perfect estimation?" This approach is limited to lossless distributed estimation. However, it provides practical insight into the more general problem. The approach presented here is closely related to the distributed source coding theorem first proved by Slepian and Wolf [90]. In fact, the distributed source coding theorem is extended to the special case when the accuracy in parameter estimation (noiseless transmission of joint-type) is the main concern, rather than perfect reconstruction of the sources per se.

For this purpose, in Section 2.4 it is shown that for optimal estimation, the *joint–type* of the correlated sequences is a sufficient statistic. Then in Section 2.5 it is shown that for any fixed source parameter, the Slepian-Wolf region is not only sufficient but also *necessary* for the perfect transmission of the joint–type, i.e., efficient estimation. A distributed coding scheme for achieving these rates is also presented. The proofs are based on the *large deviation theory* (particularly Sanov's theorem) and a particular form of distributed random binning scheme. For contrasting the contribution of this chapter, we show in Section 2.8 that the method of Han and Amari gives a set of *sufficient* rates for efficient estimation in distributed estimation (for fixed set of parameters).

As we show in the last part of the chapter, determination of the region of achievable rates for efficient estimation of a general source is an extremely difficult problem. This fact is the motivation for proposing methods that provide practical guidelines

for designing distributed estimation systems. One example of such approaches proposed in [88] for binary symmetric sources. Given any source parameter for binary symmetric source, this theorem determines the region of achievable rates for efficient estimation of the source.

In Section 2.7, we generalize this theorem for a larger class of sources. More specifically, we provide a lower bound on the region of achievable rates (i.e. existence of encoder/decoders for attaining an accuracy equivalent to local estimation) for estimation of sources with a convex mutual information with respect to the unknown parameter θ.

Finally, we study a famous *"Modulo-two adder" source network* due to [61]. We show that in such a network, if communications is for the purpose of estimation, in contrast to the Slepian–Wolf rates, the triple set of $(0, 0, 0)$ rates are achievable.

Remark 2 *The second theorem proved in this chapter (Theorem 2.5.1) shows that in a distributed coding scenario, the transmission of the joint–type needs as much rate as the transmission of the sequences themselves, i.e. the SW rates. This result was previously published in [2] and [43]. More specifically, Ahlswede and Csiszar showed that in a distributed coding framework, perfect transmission of a particular class of functions (*sensitive functions*) of the sequences can be as difficult as perfect transmission of the sequences themselves. They also showed that for the special case of the* joint-type, *the sufficient and necessary rates coincides with the SW rates. Therefore, the proof provided here can be considered as a* simple alternative proof *for this problem.*

2.2 Problem Statement

Suppose a phenomenon is characterized by a *discrete* probability distribution $Q(X, Y; \theta)$, where the random variables X and Y choose discrete values from sets of discrete alphabets $\mathcal{X}$ and $\mathcal{Y}$, respectively and where $\theta \in \Theta \subset \mathcal{R}^k$ is a parameter vector by which the PD is uniquely identified. Let also (x^n, y^n) be n *i.i.d* samples drawn from $Q(X, Y; \theta)$. The sequences are encoded into separate messages by encoding functions $f : \mathcal{X}^n \rightarrow \Pi_f$ and $g : \mathcal{Y}^n \rightarrow \Pi_g$, respectively. The message sets Π_f and Π_g, also referred to as *codebooks*, are arbitrary discrete sets $\Pi_f = \{1, 2, ..., 2^{nR_x}\}$ and $\Pi_g = \{1, 2, ..., 2^{nR_y}\}$. Here R_x and R_y are called the *code rates* defined as $R_x = \lim_{n\rightarrow\infty} \frac{\log|\Pi_f|}{n}$ and $R_y = \lim_{n\rightarrow\infty} \frac{\log|\Pi_g|}{n}$ where $|\Pi_f|$ and $|\Pi_g|$ are cardinality of the sets Π_f and Π_g, respectively.

The encoded messages are transmitted to a common receiver via separate communication channels. The communications channel capacities are generally limited. Thus the design of the codebooks and the messages involve compression. By appropriately incorporating compression such that the rate of the codes is "no more" than the available capacities of the channels, the decoder receives noiseless encoded messages. At the receiver, by means of a decoder/estimator function $h : \Pi_f \times \Pi_g \rightarrow \Theta$, it is desired to estimate the parameter of the underlying distribution, $\hat{\theta} = h(f(x^n), g(y^n))$.

The pair of rates (R_x, R_y) is called *achievable* if there exists at least one pair of encoders (f, g) and a decoder h with probability converging to 1 by which one can construct the sequences of codes that provide transmission of *sufficient statistics* for the parameter θ to the receiver with probability of error converging to 0 as n becomes sufficiently large. Here we study the region of achievable rates.

Remark 3 *Suppose X is a discrete set with finite or countably infinite cardinality. A probability distribution $Q(X; \theta)$ on $\mathcal{X}$ is defined as a function $Q : \mathcal{X} \rightarrow \mathcal{R}$ which*

satisfies:

$$Q(x) \geq 0 \quad (\forall x \in \mathcal{X}) \qquad \textit{and} \qquad \sum_{x \in \mathcal{X}} Q(x) = 1.$$

Similarly, a joint probability distribution $Q(X, Y; \theta)$ is defined similarly on two discrete sets $\mathcal{X}$ and $\mathcal{Y}$ as a function $Q : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{R}$ which satisfies:

$$Q(x, y) \geq 0 \quad (\forall (x, y) \in \mathcal{X} \times \mathcal{Y}) \qquad \textit{and} \qquad \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} Q(x, y) = 1.$$

Throughout, when clear from the context, we avoid representing the dependence of the PD $Q(X, Y; \theta)$ on the parameter θ explicitly.

2.3 Information theory- some definitions and theorems

The fundamental concept in Shannon's information theory is the typicality of large sequences. Sufficiently long sequences fall into two major sets, the typical and non-typical sets. It was proved by Shannon [87] that almost all (with probability close to 1) randomly generated sequences are members of the so-called *typical set.* The important property for all members of the typical set is that their entropy is almost equal to the entropy of the source and therefore all these sequences carry the same amount of information about the source. In other words, for sufficiently large n, the average of the negative logarithm of the probability of each sequence x^n approaches the source entropy $H(x)$. Therefore, it is a common practice in information theory to treat long sequences such that there are almost $2^{nH(X)}$ sequences each with probability of $2^{-nH(X)}$. These two observations that almost all long sequences are typical and so carry the same amount of information about the source, and that there are almost $2^{nH(X)}$ such typical sequences lead to Shannon's *source coding theorem.* This shows a code with a *rate* at least equal to the entropy of the source exists which carries

carry all the source information. Now, consider two correlated sources generating two correlated random variables X and Y with entropies $H(X)$ and $H(Y)$, respectively. Similarly there are $2^{n(H(X)+H(Y))}$ total typical sequences. Also since the variables are correlated, there are almost $2^{nH(X,Y)}$ *jointly typical sequences.*

The *method of types* [29] is a topic in combinatorics used extensively for proving theorems in information theory. This method is a set of tools for studying the behavior of long sequences using their *types.* The type of a sequence is the relative occurrence of the alphabets in the sequence. The method of types owes its usefulness (particulary in information theory) to the concept of typicality, firstly recognized by Shannon all typical sequences have the same type. Here, we present two important results from the method of types, i.e. the number of types for long sequences is upper bounded by a polynomial function of the sequence length, and the maximum number of sequences with the same type (joint-type) is upper bounded by an exponential function of the sequence length and entropy (joint entropy).

The *Large deviation theory* is another subject that is reviewed in this section. This theory involves rarely occurring events, i.e. events that are far from expectation. We present one of the most important theorems in this theory, referred to as *Sanov's theorem* that gives the likelihood of rare sequences (events) in terms of the Kullback-Leibler (KL) distance between their types and the underlying probability distribution. We will use this theorem to prove another important theorem that computes the likelihood of the event that two sequences generated independently according to two independent marginal distributions *appear* to be *jointly-typical* with respect to a joint distribution. This theorem will be the main tool used to implement the random binning scheme for transmission of the joint-type in a distributed coding scenario.

2.3.1 A quick review of fundamentals

2.3.1.1 Typical and jointly typical sets [24]

A *typical set* A_X with respect to $Q(X)$ is a set of sequences $(x_1, x_2, ..., x_n) \in \mathcal{X}^n$ with the following property:

$$2^{-n(H(X)+\epsilon)} \leq Q(x_1, x_2, ..., x_n) \leq 2^{-n(H(X)-\epsilon)}.$$

Thus, for arbitrarily small $\epsilon > 0$ and sufficiently large n:

$$A_X = \{x^n : \left| -\frac{1}{n} \log Q(x^n) - H(X) \right| \leq \epsilon\}$$

where $|.|$ denotes absolute value.

Lemma 2.3.1 *(The Asymptotic Equipartition Property, (AEP)[24]) If x_1, x_2, ... are i.i.d $\sim Q(X)$, then:*

1. $-\frac{1}{n} \log Q(x_1, x_2, ..., x_n) \rightarrow H(X)$ *in probability.*
2. *if* $(x_1, x_2, ..., x_n) \in A_X$ *then* $H(X) - \epsilon \leq -\frac{1}{n} \log Q(x_1, x_2, ..., x_n) \leq H(X) + \epsilon$.
3. $\Pr\{A_X\} > 1 - \epsilon$ *for n sufficiently large.*
4. $|A_X| \leq 2^{n(H(X)+\epsilon)}$,
5. $|A_X| \geq (1 - \epsilon)2^{n(H(X)-\epsilon)}$.

where $|.|$ denotes cardinality [1].

Proof 2.3.1 *The first statement is the fundamental concept upon which the information theory is built and is an extension of the* weak law of large numbers. *For the proof refer to [24], page 51-52.*

[1] *We assume that the meaning of notations for absolute value and cardinality are clear from the context.*

Similarly, for two correlated random variables, for arbitrarily small $\epsilon > 0$ and sufficiently large n the *jointly typical set* A_{XY} of pairs of sequences $(x^n, y^n) \sim Q(X, Y)$ is defined as:

$$\begin{aligned} A_{XY} = \{(x^n, y^n) : \left| -\frac{1}{n} \log Q(x^n) - H(X) \right| &\leq \epsilon \\ \left| -\frac{1}{n} \log Q(y^n) - H(Y) \right| &\leq \epsilon \\ \left| -\frac{1}{n} \log Q(x^n, y^n) - H(X,Y) \right| &\leq \epsilon \}. \end{aligned}$$

Lemma 2.3.2 *(The Joint Asymptotic Equipartition Property (JAEP) [24]) Let (x^n, y^n) be sequences of length n drawn i.i.d according to $Q(X, Y)$. Then:*

1. *$Pr(x^n, y^n) \in A_{XY} \rightarrow 1$ as $n \rightarrow \infty$,*
2. *$|A_{XY}| \leq 2^{n(H(X,Y)+\epsilon)}$,*
3. *If $(\tilde{x}^n, \tilde{y}^n) \sim Q(x^n)Q(y^n)$, i.e. $\tilde{x}^n$ and $\tilde{y}^n$ are generated independently by the same marginals of $Q(X, Y)$, then for sufficiently large n:*

$$\Pr\left[(\tilde{x}^n, \tilde{y}^n) \in A_{XY}\right] \leq 2^{-n(I(X;Y)-3\epsilon)},$$

Also, for sufficiently large n,

$$\Pr\left[(\tilde{x}^n, \tilde{y}^n) \in A_{XY}\right] \geq 2^{-n(I(X;Y)+3\epsilon)}.$$

Proof 2.3.2 *c.f. [24], page 195.*

Lemma 2.3.3 *Let (x^n, y^n) be sequences of length n drawn* i.i.d *according to $Q(X, Y)$. Let for any $\epsilon > 0$ define $A_X(X^n|y^n)$ to be the set of x^n sequences that are jointly typical with a particular y^n sequence. If $y^n \in A_Y$, then for sufficiently large n, we have:*

1. *$|A_X(X^n|y^n)| \leq 2^{n(H(X|Y)+2\epsilon)}$,*

2. $(1-\epsilon)2^{n(H(X|Y)-2\epsilon)} \leq \sum_{y^n} Q(y^n)|A_X(X^n|y^n)|$.

Proof 2.3.3 *See [24], page 387.*

2.3.1.2 Marginal and joint types

Let $N(a|x^n)$ be the number of occurrences of the alphabet symbol $a \in \mathcal{X}$ in sequence x^n. The *type* of the sequence x^n (or marginal type with respect to the joint distribution $Q(X,Y)$) is defined as the relative occurrence of symbol $a \in \mathcal{X}$ in the sequence:

$$\tilde{P}_{x^n}(a) = \frac{1}{n}N(a|x^n). \tag{2.1}$$

Similarly, the *joint-type* of the pair (x^n, y^n) is defined:

$$\tilde{P}_{x^n y^n}(a,b) = \frac{1}{n}N((a,b)|(x^n,y^n)). \tag{2.2}$$

2.3.1.3 Set of types ($\mathcal{P}_n$)

The set of *types with denominator* n are represented by $\mathcal{P}_n$. For example, the set of joint-types with denominator n for a binary set $\mathcal{X} \times \mathcal{Y}$ is:

$$\mathcal{P}_n = \left\{ \begin{bmatrix} P(0,0) & P(0,1) \\ P(1,0) & P(1,1) \end{bmatrix} : \begin{pmatrix} \frac{0}{n} & \frac{0}{n} \\ \frac{0}{n} & \frac{n}{n} \end{pmatrix}, \begin{pmatrix} \frac{1}{n} & \frac{0}{n} \\ \frac{0}{n} & \frac{n-1}{n} \end{pmatrix}, \ldots, \begin{pmatrix} \frac{n}{n} & \frac{0}{n} \\ \frac{0}{n} & \frac{0}{n} \end{pmatrix} \right\}.$$

Lemma 2.3.4 *For random variables X and Y selected from discrete alphabets $\mathcal{X}$ and $\mathcal{Y}$, respectively, the maximum number of joint-types with denominator n is polynomial in n:*

$$|\mathcal{P}_n| \leq (n+1)^{|\mathcal{X}\times\mathcal{Y}|}. \tag{2.3}$$

Proof 2.3.4 *Refer to ([24], page 280).*

Corollary 2.3.5 *For a binary alphabet $\mathcal{X} = \mathcal{Y} = \{0,1\}$, the cardinality of the joint-type set with denominator n is at most polynomial in n:*

$$|\mathcal{P}_n| \leq (n+1)^4$$

Remark 4 *This result provides the possibility of "almost zero–rate" transmission of the marginal types to the decoder as will be discussed later in Section 2.5.*

2.3.1.4 Type class

For any type $P_x \in \mathcal{P}_n$, the *type class*, $T(P_x)$, is the set of sequences with type equal to P_x:

$$T(P_x) = \{x^n \in \mathcal{X}^n : \tilde{P}_{x^n} = P_x\}. \tag{2.4}$$

Similarly, for any joint type $P_{xy} \in \mathcal{P}_n$ the type class, $T(P_{xy})$, is defined as the set of pairs of sequences with joint-type equal to P_{xy}:

$$T(P_{xy}) = \{(x^n, y^n) \in \mathcal{X}^n \times \mathcal{Y}^n : \tilde{P}_{x^n y^n} = P_{xy}\}. \tag{2.5}$$

Lemma 2.3.6 *For any type $\tilde{P} \in \mathcal{P}_n$:*

$$\frac{1}{(n+1)^{|\mathcal{X}\times\mathcal{Y}|}} 2^{nH(\tilde{P})} \leq |T(\tilde{P})| \leq 2^{nH(\tilde{P})} \tag{2.6}$$

where $H(\tilde{P})$ is defined below.

Proof 2.3.5 *Refer to ([24] page 282).*

Remark 5 *The important result of this section is that while there exists only a polynomial number of types with denominator n, e.g. $(n+1)^4$ for binary sequences, there exists an exponential number of sequences with the same type. This property shows that there should exist at least "one type class" with an exponential number of sequences, i.e. the type class with entropy $H(X)$. We use this result in distributed coding scheme in later sections.*

2.3.1.5 Entropy of a type

The entropy of a type $\tilde{P}_{x^n}$ is defined as:

$$H(\tilde{P}_{x^n}) \triangleq -\sum_{a\in\{0,1\}} \tilde{P}_{x^n}(a) \log \tilde{P}_{x^n}(a). \tag{2.7}$$

Similarly, the entropy of a joint–type is defined:

$$H(\tilde{P}_{x^n y^n}) \triangleq -\sum_{x^n, y^n} \tilde{P}_{x^n y^n} \log \tilde{P}_{x^n y^n}. \tag{2.8}$$

2.3.2 Large deviation theory

Typical sequences are the main building blocks of information theory. As the sequences become larger, the probability of a typical sequence occurring approaches 1. In contrast, large deviation theory studies the probability of the occurrence of *non-typical sequences.* In other words, this theory is concerned with events that occur far from the expectation. For instance, in a sequence of coin flips, the likelihood (probability) of observing a sequence with almost the same number of "head"s and "tail"s is close to 1. In contrast, as it is shown by large deviation theory, the likelihood of observing a sequence that has 9 times more "head"s (one's) than it has "tails"s (zero's) is exponentially low and a function of the KL–distance between the underlying probability distribution (in this case uniform) with the *type* of the observed sequence (i.e. $(\frac{1}{10}, \frac{9}{10})$).

Example ([24], page 291): What is the probability that $\frac{1}{n}\sum x_i$ is near $\frac{1}{3}$, if $x_1, x_2, ..., x_n$ are drawn i.i.d Bernoulli($\frac{1}{3}$)? Since the empirical mean is expected to be close to the expected value ($\frac{1}{3}$) this is an event with probability close to 1 and so a small deviation from the expectation. On the other hand, suppose we are interested in the probability of the event when $\frac{1}{n}\sum x_i$ is greater than $\frac{3}{4}$? The probability of this event is exponentially small and so is a large deviation from the expectation. Using large deviation theory, the probability of this event turns out to be $2^{-nD(\tilde{P}_{x^n}\|Q(X))} = 2^{-nD((\frac{1}{4},\frac{3}{4})\|(\frac{2}{3},\frac{1}{3}))}$, where $\tilde{P}_{x^n} = (N(0|x^n), N(1|x^n)) = (\frac{1}{4}, \frac{3}{4})$ is the type of observations $x_1, x_2, ..., x_n$, i.e. $\frac{1}{n}\sum x_i = \frac{3}{4}$. The notation $N(a|x^n)$ defines the relative occurrence of alphabet a in x^n as defined previously in (2.2). Also $Q = (\Pr(X = 0), \Pr(X = 1)) = (\frac{2}{3}, \frac{1}{3})$ is the probability distribution from which the

variables are drawn.

Now let E be a subset of the set of probability mass functions. We write:

$$\Pr_Q(E) \triangleq Q(E) = Q(E \cap \mathcal{P}_n) = \sum_{x^n : \tilde{P}_{x^n} \in (E \cap \mathcal{P}_n)} Q(x^n), \tag{2.9}$$

which is in fact the probability under distribution Q of the sequences whose type is in E with respect to PD $Q(X)$. Here $\Pr_Q$ stands for the probability with respect to $Q(X)$ [2]. *Sanov's Theorem* provides an upper bound on the likelihood of these sequences as well as a method for computing the best probability distribution in E that maximizes this likelihood.

Theorem 2.3.7 *(Sanov's Theorem [24]) Let $x_1, ..., x_n$ be i.i.d $\sim Q(X)$. Let $E \subseteq \mathcal{P}_n$ be a set of probability distributions. Then:*

$$Q(E) = Q(E \cap \mathcal{P}_n) \leq (n+1)^{|\mathcal{X}|} 2^{-nD(P^* \| Q)}, \tag{2.10}$$

where

$$P^* = \arg\min_{P \in E} D(P \parallel Q), \tag{2.11}$$

is the distribution in E that is closest to Q in relative entropy (K–L distance). If in addition, the set E is the closure of its interior, then:

$$\frac{1}{n} \log Q(E) \rightarrow -D(P^* \parallel Q). \tag{2.12}$$

Proof 2.3.6 *The proof is based on the* method of types. *For detail refer to ([24], page 292).*

In the following section, Sanov's Theorem is used to derive an important property of independently generated sequences. This property will be used in the proposed *distributed binning scheme* presented in later sections.

[2] We follow classical notations used in Thomas and Cover [24]

Remark 6 *The following theorem is the basis for the distributed coding scheme presented in the following sections. The proof is based on the following observation:*

Consider a product distribution $Q_0(X,Y) = Q(X)Q(Y)$ consisting of the product of the marginals of a joint distribution $Q(X,Y)$. In series of n experiments measuring the output of this source (Q_0), the event of observing a pair (x^n, y^n) that are statistically independent (because their PD is the product of its marginals) is a typical event-its probability or likelihood is close to 1. *In contrast, the event of observing a pair that is jointly typical with respect to the joint distribution $Q(X,Y)$ is a rare (non–typical) event. The following theorem is concerned with measuring the probability of observing such events. It is proved in the following theorems that the probability of such events is exponentially low with exponent proportional to the* mutual information *between the correlated random variables.*

Moreover, we show that the probability of observing pairs with a joint–type that has entropy less than the entropy of the joint–type $Q(X,Y)$ is also exponentially low with exponent proportional to the mutual information. This idea provides the theoretical basis for using the minimum entropy *decoder; a decoder that outputs a pair with minimum joint–type entropy. This result will be used in our main distributed coding scheme explained in detail in Section 2.5.1.*

Lemma 2.3.8 *(Mutual Dependence Theorem)*

Part (a)([24], page 295): Let $Q(X,Y)$ be a given joint distribution and let $Q_0(X,Y) = Q(X)Q(Y)$ be the associated product distribution formed from the marginals of Q. Let $\mathcal{E}_0$ be the event that a sample drawn according to Q_0 "appears" to be jointly distributed according to Q, i.e. has a joint–type Q. The probability of this rare event is:

$$\Pr(\mathcal{E}_0) \leq 2^{-nI(X,Y)}, \tag{2.13}$$

where $I(X;Y)$ is the mutual information corresponding to $Q(X,Y)$. Moreover, the joint-type of the most probable pair that achieves the equality *is equal to the* joint

distribution:

$$\tilde{P}^*_{x^n y^n} = Q(X,Y). \tag{2.14}$$

Part (b): Let $\mathcal{E}_u$ be the event that a sample drawn according to Q_0 "appears" to be jointly distributed with respect to any other joint distribution *$Q_u \neq Q$ for which $H(Q_u) \leq H(Q)$. The probability of this event is:*

$$\Pr(\mathcal{E}_u) \leq 2^{-nI(X,Y)}. \tag{2.15}$$

The joint-type of the most probable pair that achieves the equality satisfies:

$$H(\tilde{P}^*_u) \leq H(Q). \tag{2.16}$$

Proof 2.3.7 *For proof of part (a) See Appendix B. For proof of part (b), denote the mutual information with respect to Q_u as $I_u(X;Y)$. First having assumed the condition $H(Q_u) \leq H(Q)$:*

$$\begin{aligned} I_u(X;Y) &= H(X) + H(Y) - H(Q_u) \\ &\geq H(X) + H(Y) - H(Q) \\ &= I(X;Y). \end{aligned}$$

Based on this observation and according to Part (a):

$$\begin{aligned} \Pr(\mathcal{E}_u) &\leq 2^{-nI_u(X,Y)} \\ &\leq 2^{-nI(X,Y)}. \end{aligned}$$

Also, since $\tilde{P}^*_u = Q_u$ and $H(Q_u) \leq H(Q)$ we conclude that $H(\tilde{P}^*_u) \leq H(Q)$.

Theorem 2.3.9 *Let $Q(X,Y)$ be a given joint distribution and let $Q_0(X,Y) = Q(X)Q(Y)$ be the associated product distribution formed from the marginals of Q. Let $\mathcal{E}$ be the event that a sample drawn according to Q_0 has joint-type Q or any other joint-type for which $H(\tilde{P}_{x^n y^n}) \leq H(Q)$. The probability of this event is:*

$$\Pr(\mathcal{E}) \leq (n+1)^{|\mathcal{X}\times\mathcal{Y}|} 2^{-nI(X,Y)}.$$

Proof 2.3.8 *By lemma (2.3.8), the probability of a sequence appearing with joint-type $Q(X,Y)$ is less than $2^{-nI(X,Y)}$. By the same lemma, the probability of a sequence appearing with any joint-type for which $H(\tilde{P}_{x^n y^n}) \leq H(Q)$ is also less than $2^{-nI(X,Y)}$. There are at most $(n+1)^{\mathcal{X}\times\mathcal{Y}}$ joint-types with denominator n. The probability of the union of these events are less than the sum of them, and therefore the theorem is proved.*

Remark 7 *This theorem shows that in a distributed random binning situation, if a pair with joint–type equal to Q is observed, the probability of another pair appearing with the* same *joint–type or* any other *joint–type with entropy less than $H(Q)$ is exponentially low with exponent proportional to the mutual information.*

Remark 8 *Suppose a pair of sequences generated by the product distribution $Q_0 = Q(X)Q(Y)$ exists in a bin that appears jointly typical with respect to $Q(X,Y)$. For this pair the joint-type $\tilde{P}_{x^n y^n} = Q(X,Y)$. This is equivalent to the case when a pair is generated by the joint distribution $Q(X,Y)$, i.e. for which case again $\tilde{P}_{x^n y^n} = Q(X,Y)$. This observation is the basis for the universal coding scheme presented in upcoming sections.*

2.4 Sufficiency of the joint–type

We will see in the following chapters that the joint–type is a sufficient statistic for ML estimation of θ using the following theorem.

Lemma 2.4.1 *Let (X^n, Y^n) be n samples drawn i.i.d from $Q(X,Y;\theta)$ with $(X,Y) \in \mathcal{X}\times\mathcal{Y}$. Then the likelihood of the data depends only on the joint type $\tilde{P}_{x^n y^n}$ is given by*

$$\mathcal{L}(\theta) = 2^{-n(H(\tilde{P}_{x^n y^n}) + D(\tilde{P}_{x^n y^n} \| Q))},$$

where D is the Kullbak-Liebler (KL) distance.

Proof 2.4.1 *[24]:*

$$\begin{aligned}
Q(x^n, y^n; \theta) &= \prod_{i=1}^{n} Q(x_i, y_i; \theta) \\
&= \prod_{(a,b)\in\mathcal{X}\times\mathcal{Y}} Q(a, b; \theta)^{N((a,b)|x,y)} \\
&= \prod_{(a,b)\in\mathcal{X}\times\mathcal{Y}} Q(a, b; \theta)^{\tilde{P}_{x^n y^n}(a,b)} \\
&= \prod_{(a,b)\in\mathcal{X}\times\mathcal{Y}} 2^{n(\tilde{P}_{x^n y^n}(a,b) \log Q(a,b;\theta))} \\
&= \prod_{(a,b)\in\mathcal{X}\times\mathcal{Y}} 2^{n(\tilde{P}_{x^n y^n}(a,b) \log Q(a,b;\theta) - \tilde{P}_{x^n y^n}(a,b) \log \tilde{P}_{x^n y^n}(a,b) + \tilde{P}_{x^n y^n}(a,b) \log \tilde{P}_{x^n y^n}(a,b))} \\
&= 2^{n \sum_{(a,b)\in\mathcal{X}\times\mathcal{Y}} (\tilde{P}_{x^n y^n}(a,b) \log \frac{\tilde{P}_{x^n y^n}(a,b)}{Q(a,b,\theta)} + \tilde{P}_{x^n y^n}(a,b) \log \tilde{P}_{x^n y^n}(a,b))} \\
&= 2^{n(-D(\tilde{P}_{x^n y^n} \| Q) - H(\tilde{P}_{x^n y^n}))}. \qquad (2.17)
\end{aligned}$$

Remark 9 *The result of this theorem is the basis for all developments presented in the following chapters when an ML estimation problem is solved. When a complete set of data (x^n, y^n) is available, the ML solution involves a maximization of the first term in Equation (2.17). On the other hand, when the data is partially available, also referred to as* incomplete–data, *e.g. when only y^n is available, ML estimation involves the maximization of a lower bound on (2.17). This will be used in the following chapters.*

We now state the main result of this section from statistics that will be used in the following sections.

Theorem 2.4.2 *(Sufficiency of the Joint Type) Let (X^n, Y^n) be n samples drawn i.i.d from $Q(X, Y; \theta)$. The joint type $\tilde{P}_{x^n y^n}$ is a sufficient statistic for θ.*

Proof 2.4.2 *Let $r(X^n, Y^n) \triangleq 2^{-n(H(\tilde{P}_{x^n y^n}))}$ and $q(S(X^n, Y^n); \theta) \triangleq 2^{nD(\tilde{P}_{x^n y^n} \| Q)}$. We observe that the joint type factorizes the likelihood. Therefore, the proof is immediate from the* Factorization Theorem *stated in Section 1.1.1.3.*

2.5 Coding for distributed estimation

The main obstacle in designing a universal coding scheme in distributed estimation is that the underlying PD $Q(X, Y; \theta)$ changes with the unknown parameter θ, leading to different correlations between the sequences, hence a different conditional type. Even when the unknown parameter is fixed, determination of the region of achievable rates is not obvious. In what follows, the region of rates for this case is studied. With a simple proof it is shown that for any fixed value for the parameter θ, the region of achievable rates coincides with the Slepian-Wolf (SW) region. The proof is based on the *large deviation theory* that provides a very simple alternative to the proofs provided previously [2].

In the following a coding scheme for proving the main theorem is presented. The proof is based on *distributed binning* [3]. The process consists of two steps. In the first step, a codebook is constructed. Each source has access to a marginal distribution of the joint distribution. Consequently, it randomly assigns all sequences generated according to its marginal distribution to a set of bins[4]. Then the bins generated by each source are put together to form joint-bins, also referred to as codebook. In the second step, the transmission of measured sequences is performed. For this purpose, given the sequence of measurements, the encoder at each source looks into all its bins and transmits the address of the bin to which the sequence belongs. At the decoder, it is desired to locate the pair in the joint-bin corresponding to the received address. For this purpose, the decoder selects a pair in the joint-bin for which the joint–entropy is minimum. We refer to this decoder as the *minimum–entropy* decoder [30].

The main idea in the proof is to show that with the aid of minimum–entropy decoder, if the rates are chosen properly (similar to the Slepian–Wolf rates for any

[3] We coin this term as a generalization to the binning scheme defined in [24]

[4] Notice that all distributions including the marginal distributions are functions of the parameter θ

fixed parameter), the probability of decoding a pair with a joint–type other than the joint–type of the measurement pair goes to zero exponentially. Conversely, it is shown that when the probability of decoding a pair with a joint–type different than the joint–type of the measurement pair vanishes, the proposed rates are also necessary.

Figure 2.1 illustrates the distributed binning scheme for a bivariate source. Each dimension corresponds to one of the sources. Source X assigns almost $2^{nH(X)}$ sequences generated according to $Q(X)$ to 2^{nR_x} bins. Similarly for source Y. Suppose a pair appears (marked by *star* $(*)$) to be jointly distributed according to the joint distribution $Q(X,Y)$. Theorem 2.3.8 shows that the probability of such a pair appearing in the bin is exponentially low (almost $2^{-nI(X,Y)}$). Moreover, Theorem 2.3.9 shows that the probability of pairs appearing with a joint–type different than the joint–type of the measurement pair is also exponentially low. Therefore, when the rates are chosen sufficiently large, there would exist sufficient diversity in bins to avoid the existence of more than one such pairs in a joint–bin. This observation is the basis for the following proofs.

2.5.1 Coding scheme

Suppose we fix the parameter θ. We will denote the PD $Q(X,Y;\theta)$ with fixed θ by $Q(X,Y)$. Here, for a two–terminal estimation scenario, the sufficient and necessary rates as well as a coding scheme are presented to noiselessly reconstruct the joint–type at the decoder and therefore efficiently estimate θ.

Distributed random binning: Suppose it is desired to transmit the joint–type of a pair of n *i.i.d.* samples (x^n, y^n) from the joint distribution $Q(X,Y)$. The sources X and Y partition the space of $\mathcal{X}^n$ into 2^{nR_x} and the space of $\mathcal{Y}^n$ into 2^{nR_y} bins, respectively.

Random code generation and binning: To form the codebooks, the sequences

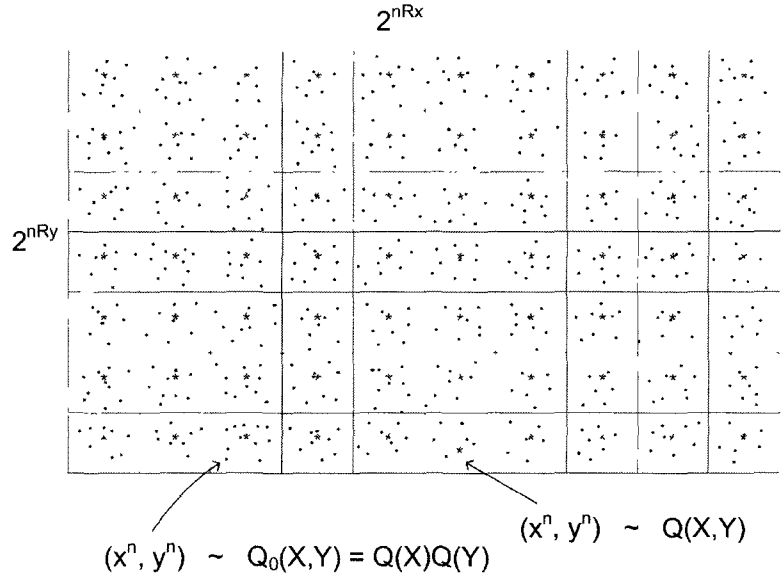


Figure 2.1: Codebook for a distributed binning scheme

x^n and y^n are generated according to the marginal distributions $Q(X)$ and $Q(Y)$, respectively. This is due to the fact that the sources do not collaborate and therefore have no information of the underlying joint distribution. The marginals are associated with the joint PD $Q(X,Y)$. The source X randomly assigns every $x^n \in \mathcal{X}^n$ to one of 2^{nR_x} bins according to a uniform distribution on $\{1, 2, ..., 2^{nR_x}\}$. Independently, the source Y randomly assigns the sequences $y^n \in \mathcal{Y}^n$ generated by marginal distribution $Q(Y)$ to the 2^{nR_y} bins $\{1, 2, ..., 2^{nR_y}\}$. We call the sequences x^n and y^n the *codewords* and the set of bins for x^n and y^n as the *codebooks* Π_f and Π_g, respectively. The assignment functions f and g are revealed to both the encoder and decoder. Note that the generation of the sequences are according to the marginal distributions rather than the joint distribution.

Encoding: For any given pair (x^n, y^n) generated by $Q(X,Y)$, the sender X sends the index m_{x^n} of the bin to which x^n belongs, to the decoder. Similarly for Y. Note

that the encoded pair (x^n, y^n) is generated by the joint distribution.

Decoding: Given the received bin index pair (m_{x^n}, m_{y^n}), a *minimum entropy decoder* $h(m_{x^n}, m_{y^n})$ declares $Q(X,Y) = \tilde{P}_{x^n y^n}$ the joint–type of a pair with minimum entropy, i.e. $H(\tilde{P}_{x^n y^n}) \leq H(\tilde{P}_{x'_n y'_n})$ for all sequences in the joint–bin $(x'_n, y'_n) \neq (x^n, y^n)$.

Probability of error P_e: In conventional decoding (e.g. the SW decoding [24], page 411) an error is declared if (x^n, y^n) is not in the joint–typical set (A_{XY}) or if there is another *jointly typical* sequence in the same bin. Instead we are interested in *just* the *joint-type* of the sequences x^n and y^n. Therefore, considering the minimum entropy decoder, an error is declared if there exists at least one sequence in the bin that has a joint–type with entropy less than $H(Q)$ but is not equal to $Q(X,Y)$.

Achievability: The rate R is called *achievable* if there exists at least one pair of encoders (f, g) and a decoder h with probability converging to 1 by which one can construct sequences of codes that provide transmission of the joint–type of the sequence (x^n, y^n) to the receiver with probability of error converging to 0 as n becomes sufficiently large.

Remark 10 *We recall Remark 7 and Theorem 2.3.9. The proof is based on this the that in a distributed random binning situation, if a pair with joint–type equal to Q is observed, the probability of another pair appearing with the* same *joint–type or* any other *joint–type with entropy less than $H(Q)$ is exponentially low with exponent proportional to the mutual information.*

We now present the main theorem.

Theorem 2.5.1 *For any fixed θ the following rates are sufficient and necessary:*

$$\begin{aligned} R_x &\geq H(X|Y;\theta), \\ R_y &\geq H(Y|X;\theta), \\ R_x + R_y &\geq H(X,Y;\theta). \end{aligned}$$

Although the achievability is an obvious result of the Slepian-Wolf theorem, here a different proof is presented. The necessity is also proved in the converse part.

Proof 2.5.1 (Achievability) *The proof is similar to the proof of Slepian-Wolf Theorem given in ([24], page 412). The codebooks are generated by two marginal distributions of the PD $Q(X,Y)$. At the decoder, the marginal codebooks are paired up together to form almost $2^{n(R_x+R_y)}$* joint–bins. *Now suppose a pair (x^n, y^n) is generated according to $Q(X,Y)$. The encoder finds the bins to which these sequences belong. On the receiver side a minimum entropy decoder is used. Therefore, according to Theorem 2.3.9 the probability of any pair appearing with joint–type other than $Q(X,Y)$ with entropy less than H(Q), is upper–bounded by $2^{-nI(X,Y)}$. On the other hand, there are almost $2^{n(H(X)+H(Y))}$ sequences. Therefore, on average, a rate larger than $2^{-nI(X,Y)}2^{n(H(X)+H(Y))} = 2^{n(H(X,Y))}$ is needed to make sure that no such pair (with joint–type other than Q(X,Y) and less entropy) appears in the given bin. It is important to notice that although the codebooks are generated by the marginal distributions, the encoded pairs are generated according to the joint distribution $Q(X,Y)$. The formal proof follows.*

Consider any pair $(x^n, y^n) \sim Q(X,Y)$. At the decoder side, it is desired to reconstruct the joint type $\tilde{P}_d$, or better, the joint distribution $Q(X,Y)$.

We define the following error events (for brevity we represent $Q(X,Y)$ by Q below):

$$E_0 = \{(x^n, y^n) \notin A_{XY}\}, \tag{2.18}$$

and

$$\begin{aligned} E_1 &= \{\exists(x'_n, y^n) : x'_n \neq x^n, f(x'_n) = f(x^n), \text{ and} \\ &\quad \tilde{P}_{x'_n y^n} \neq Q \text{ for which } H(\tilde{P}_{x'_n y^n}) \leq H(Q)\}, \end{aligned} \tag{2.19}$$

.

The error E_1 is declared when there exists a pair in the bin that has a joint-type other than $Q(X,Y)$ and whose entropy is less than the entropy $H(Q)$. This error is associated with the minimum–entropy *decoder. We showed in Theorem 2.3.9 that the probability of this event is exponentially low with exponent proportional to the mutual information. This is going to be used in the following proofs. The errors E_2 and E_3 are defined similarly.*

$$\begin{aligned} E_2 &= \{\exists(x^n, y'_n) : y'_n \neq y^n, g(y'_n) = g(y^n), \text{ and} \\ &\quad \tilde{P}_{x^n y'_n} \neq Q \text{ for which } H(\tilde{P}_{x^n y'_n}) \leq H(Q)\}, \end{aligned} \tag{2.20}$$

$$\begin{aligned} E_3 &= \{\exists(x'_n, y'_n) : (x'_n, y'_n) \neq (x^n, y^n), f(x'_n) = f(x^n), g(y'_n) = g(y^n), \text{ and} \\ &\quad \tilde{P}_{x'_n y'_n} \neq Q \text{ for which } H(\tilde{P}_{x'_n y'_n}) \leq H(Q)\}. \end{aligned} \tag{2.21}$$

The total probability of error is given by the union of the events:

$$P_e = P(E_0 \cup E_1 \cup E_2 \cup E_3) \leq P(E_0) + P(E_1) + P(E_2) + P(E_3).$$

We now study the error events. The encoded pair (x^n, y^n) is generated according to the joint PD $Q(X,Y)$. By the joint asymptotic equipartition property *([24] page 385), $P(E_0) \to 0$ and hence for n sufficiently large and arbitrary $\epsilon > 0$, $P(E_0) < \epsilon$.*

For all possible pairs $(x^n, y^n) \sim Q(X,Y)$, *the* $P(E_1)$ *can be bound as follows:*

$$\begin{aligned}
P(E_1) &= P\{\exists(x'_n, y^n) : x'_n \neq x^n, f(x'_n) = f(x^n), x'_n \in A_X \text{ and} \\
&\quad \tilde{P}_{x'_n y^n} \neq Q \text{ for which } H(\tilde{P}_{x'_n y^n}) \leq H(Q)\} \\
&= \sum_{(x^n,y^n)} p(x^n, y^n) \\
&\quad P\{\exists(x'_n, y^n) : x'_n \neq x^n, f(x'_n) = f(x^n), x'_n \in A_X \text{ and} \\
&\quad \tilde{P}_{x'_n y^n} \neq Q \text{ for which } H(\tilde{P}_{x'_n y^n}) \leq H(Q)\} \\
&\leq \sum_{(x^n,y^n)} p(x^n, y^n) \sum_{(x'_n, y^n) \neq (x^n, y^n), x'_n \in A_X} \\
&\quad P\{f(x'_n) = f(x^n)\} P\left\{\tilde{P}_{x'_n y^n} \neq Q \text{ for which } H(\tilde{P}_{x'_n y^n}) \leq H(Q)\right\} \\
&\leq \sum_{(x^n,y^n)} p(x^n, y^n) 2^{-nR_x} (n+1)^{\mathcal{X}\times\mathcal{Y}} 2^{-nI(X,Y)} |A_X| \\
&\leq (n+1)^{\mathcal{X}\times\mathcal{Y}} 2^{-nR_x} 2^{-nI(X,Y)} 2^{n(H(X)+\epsilon)} \qquad (2.22) \\
&= (n+1)^{\mathcal{X}\times\mathcal{Y}} 2^{-nR_x} 2^{n(H(X|Y)+\epsilon)}.
\end{aligned}$$

where we used part (1) of Lemma 2.3.9 (Section 2.3) in (2.22). Also we used the fact that $\sum p(x^n, y^n)|A_X| \leq 2^{n(H(X)+\epsilon)}$. *This bound goes to* 0 *if* $R_x > H(X|Y) + \epsilon$. *Notice that the rate of increase of* $(n+1)$ *is polynomial compared to the exponential decrease of the remaining terms. Hence for sufficiently large* n, $P(E_1) < \epsilon$. *Similarly for* E_2 *and* E_3. *Since the average probability of error is less than* 4ϵ, *there exists at least one code (f, g, h) with the probability of error less than* ϵ *for which the probability of error* $P_\epsilon^{(n)} \to 0$. *Thus we can reconstruct the joint-type of original sequences by a minimum entropy decoder that outputs the sequence with a joint-type with minimum entropy. This proves that the pair of sequences* (x^n, y^n) *with the same average joint–type information are distinguishable at the decoder with arbitrarily small error. This guarantees, on average, that the joint–type of the pairs are preserved and therefore the region is achievable. Since the joint–type is a sufficient statistics for* θ, *we see that*

the parameter θ can be estimated from this pair.

Remark 11 *It is mentioned previously that for estimation of any fixed parameter θ, the Slepian–Wolf rates are sufficient. The converse part of this theorem proves that these rates are also* necessary. *In other words, even when transmission is not for the purpose of perfect reconstruction of sequences but rather for estimation of the source parameters, the same rates are necessary.*

2.6 Proof of the converse

The proof to the converse is more involved and requires a preliminary theorem that relates the degree of randomness of the decoded pair with the probability of error. The theorem plays the same role in distributed coding as the Fano's inequality plays in coding.

Theorem 2.6.1 *(Extension of Fano's Inequality to Distributed Coding) Consider the decoder in the distributed binning scheme when the index of bins $I_0 \in \{1, 2, ..., 2^{nR_x}\}$ and $J_0 \in \{1, 2, ..., 2^{nR_y}\}$ are given. We define the event of error in a minimum entropy decoder as follows:*

$$E = \begin{cases} 1, & H(\tilde{P}_{x^n y^n}) \leq H(Q), \tilde{P}_{x^n y^n} \neq Q(X, Y); \\ 0, & \textit{otherwise.} \end{cases}$$

where we denote $\Pr(E = 1) = P_e$. *We have:*

$$\begin{aligned} H(X^n|I_0, J_0, Y^n) &\leq 1 + nP_e[-R_x + H(X|Y) \\ &\quad + \frac{c}{n}\log(n+1)], \\ H(Y^n|I_0, J_0, X^n) &\leq 1 + nP_e[-R_y + H(Y|X) \\ &\quad + \frac{c}{n}\log(n+1)]. \end{aligned}$$

Similarly:

$$\begin{aligned} H(X^n, Y^n | I_0, J_0) &\leq 1 + nP_e[-R_x - R_y \\ &\quad + H(X,Y) + \frac{c}{n}\log(n+1)], \end{aligned}$$

where we defined the constant $c = |\mathcal{X}|.|\mathcal{Y}|$.

Proof 2.6.1 *We first study the corner point of the rate region with pair of rates* $(H(X|Y), H(Y))$*, i.e.* Y^n *is completely given at the decoder. Following the same path as the proof of* Fano's inequality *([24], page 39) we would like to compute* $H(X^n|I_0, J_0, Y^n)$*. Let us write* $H(E, X^n|I_0, J_0, Y^n)$ *in two different forms:*

$$\begin{aligned} H(E, X^n | I_0, J_0, Y^n) &= H(X^n | I_0, J_0, Y^n) + H(E | I_0, J_0, X^n, Y^n) \\ &= H(E | I_0, J_0, Y^n) + H(X^n | E, I_0, J_0, Y^n). \end{aligned} \tag{2.23}$$

To proceed, we need to prove the following lemma:

Lemma 2.6.2

$$\begin{aligned} (a)\ H(X^n | I_0, J_0, Y^n, E=1) &\leq n(-R_x + H(X|Y) + \frac{c}{n}\log(n+1)), \\ (b)\ H(Y^n | I_0, J_0, X^n, E=1) &\leq n(-R_y + H(Y|X) + \frac{c}{n}\log(n+1)), \\ (c)\ H(X^n, Y^n | I_0, J_0, E=1) &\leq n(-R_x - R_y + H(X,Y) + \frac{c}{n}\log(n+1)). \end{aligned}$$

Proof: (a) Given Y^n *and the index of the joint bin* (I_0, J_0) *and considering the fact an error has occurred* $(E = 1)$*, the conditional entropy of* X^n *can be upper bounded by the* log *of the number of possible outcomes of the event that a sequence* X^n *in the same bin appears to* look *jointly typical with* Y^n *with respect to any joint distribution other than* $Q(X,Y)$*. The probability of such a sequence* X^n *appearing in the bin* (I_0, J_0) *that also looks jointly typical with* Y^n*, has a joint-type with entropy less than the entropy of the joint distribution, i.e.* $H(\tilde{P}_{x^n y^n}) \leq H(Q)$*, and has a joint-type not equal to the* $Q(X,Y)$ *is less than* $2^{-R_x}(n+1)^c 2^{-n(I(X;Y)+\epsilon)}$ *(see Theorem*

2.3.9). The total number of typical sequences X^n is almost $2^{n(H(X)+\epsilon)}$. Therefore, assuming uniform sampling, the total number of sequences will be less than $(n+1)^c 2^{-nR_x} 2^{nH(X)} 2^{-nI(X,Y)}$ and therefore the conditional entropy is upper bounded by $\log\left[(n+1)^c 2^{-nR_x} 2^{nH(X)} 2^{-nI(X,Y)}\right]$ and the proof is complete. □

Now we return to Equation 2.23. Observe that $H(E|I_0, J_0, X^n, Y^n) = 0$ and $H(E|I_0, J_0, Y^n) \leq H(E) = H(P_e) \leq 1$. Also from the above lemma $H(X^n|E, I_0, J_0, Y^n) \leq n(-R_x + H(X|Y) + \frac{c}{n}\log(n+1))$. Thus we have:

$$\begin{aligned}
H(X^n|I_0, J_0, Y^n) &= H(E|I_0, J_0, Y^n) + H(X^n|E, I_0, J_0, Y^n) \\
&= H(E|I_0, J_0, Y^n) \\
&+ \Pr(E=0)H(X^n|E=0, I_0, J_0, Y^n) \\
&+ \Pr(E=1)H(X^n|E=1, I_0, J_0, Y^n) \\
&\leq 1 + (1-P_e)0 + P_e n(-R_x + H(X|Y) + \frac{c}{n}\log(n+1)) \\
&= 1 + nP_e(-R_x + H(X|Y) + \frac{c}{n}\log(n+1)),
\end{aligned}$$

and therefore the proof for the first part is complete. The proof for the other corner point of the region, i.e. $(H(Y|X), H(X))$ is similarly done by exchanging the roles of x and y in the above proof. For the non-corner points of the region, one can use a similar approach and part (c) of the lemma to show that:

$$\begin{aligned}
H(X^n, Y^n|I_0, J_0) &= H(E|I_0, J_0, X^n, Y^n) + H(X^n, Y^n|E, I_0, J_0) \\
&= H(E|I_0, J_0, , X^n, Y^n) \\
&+ \Pr(E=0)H(X^n, Y^n|E=0, I_0, J_0) \\
&+ \Pr(E=1)H(X^n, Y^n|E=1, I_0, J_0) \\
&\leq 1 + (1-P_e)0 + P_e n(-R_x - R_y + H(X,Y) + \frac{c}{n}\log(n+1)) \\
&= 1 + nP_e(-R_x - R_y + H(X,Y) + \frac{c}{n}\log(n+1)),
\end{aligned}$$

□

2.6.1 Proof of the converse to Theorem 2.5.1 (Necessary condition)

We now proceed to complete the proof for the converse part of the Theorem 2.5.1. We follow the same steps as in ([24], page 413). Let f,g, and h be fixed, and $I_0 = f(X^n)$ *and* $J_0 = g(Y^n)$. *According to Theorem 2.6.1 we have:*

$$\begin{aligned} H(X^n, Y^n | I_0, J_0) &\leq 1 + n\epsilon_n^1, \\ H(X^n | I_0, J_0, Y^n) &\leq 1 + n\epsilon_n^2, \\ H(Y^n | I_0, J_0, X^n) &\leq 1 + n\epsilon_n^3, \end{aligned}$$

where we defined $\epsilon_n^1 = P_e(-R_x - R_y + H(X,Y) + \frac{c}{n}\log(n+1))$, $\epsilon_n^2 = P_e(-R_x + H(X|Y) + \frac{c}{n}\log(n+1))$ *and* $\epsilon_n^3 = P_e(-R_y + H(Y|X) + \frac{c}{n}\log(n+1))$. *The probability error* P_e *is defined previously in (2.5.1) and (2.6.1). For the converse proof we assume that when* $n \to \infty$, *the probability of error goes to zero* ($P_e \to 0$) *and therefore* $\epsilon_n^1, \epsilon_n^2, \epsilon_n^3 \to 0$. *It is desired that the rates* R_x, R_y, $R_x + R_y$ *are necessarily greater than* $H(X)$, $H(Y)$, *and* $H(X,Y)$, *respectively.*

Notice that as n is increased, the fraction $\frac{\log n}{n}$ *vanishes. We have:*

$$\begin{aligned} n(R_x + R_y) &\geq H(I_0, J_0) \\ &= I(X^n, Y^n; I_0, J_0) + H(I_0, J_0 | X^n, Y^n) \\ &= I(X^n, Y^n; I_0, J_0) \\ &= H(X^n, Y^n) - H(X^n, Y^n | I_0, J_0) \\ &\geq H(X^n, Y^n) - (1 + n\epsilon_n^1) \\ &= nH(X,Y) - n(\frac{1}{n} + \epsilon_n^1). \end{aligned}$$

Also,

$$\begin{aligned} n(R_x) &\geq H(I_0) \\ &\geq H(I_0|Y^n) \\ &= I(X^n; I_0|Y^n) + H(I_0|X^n, Y^n) \\ &= I(X^n; I_0|Y^n) \\ &= H(X^n|Y^n) - H(X^n|I_0, J_0, Y^n) \\ &\geq H(X^n|Y^n) - (1 + n\epsilon_n^2) = nH(X|Y) - n(\frac{1}{n} + \epsilon_n^2), \end{aligned}$$

and similarly

$$n(R_y) \geq nH(Y|X) - n(\frac{1}{n} + \epsilon_n^3).$$

Dividing these inequalities by n and taking the limit as $n \to \infty$, we have the desired converse proof.

2.7 On the region of achievable rates

As we pointed out previously, determination of the region of achievable rates for efficient estimation of a general source is an extremely difficult problem. This fact is the motivation for proposing methods that provide practical guidelines for designing distributed estimation systems. One example of such approaches proposed in [88] for binary symmetric sources. In this section, we generalize this theorem for a larger class of sources. More specifically, we provide a lower bound on the region of achievable rates (i.e. existence of encoder/decoders for attaining an accuracy equivalent to local estimation) for estimation of sources with a convex mutual information with respect to the unknown parameter θ.

We first present the Amari's method beginning with a definition.

Binary symmetric source: The binary symmetric source (BSS) with parameter θ is a double binary source with the probability distribution defined as the following:

$$p(X,Y;\theta) = \begin{pmatrix} \frac{\theta}{2} & \frac{1-\theta}{2} \\ \frac{1-\theta}{2} & \frac{\theta}{2} \end{pmatrix},$$

where $0 \leq \theta < \frac{1}{2}$.

Theorem 2.7.1 *We assume that the parametric family $P(X,Y;\theta)$ is defined in the region $0 < \theta < \theta'$ or $1-\theta' < \theta < 0$, where $0 < \theta' < \frac{1}{2}$. If $R \geq H_b(\theta')$, θ can be estimated without loss of information, that is, we can attain the same variance as when both x^n and y^n can be observed [88].*

Given any parameter value θ', this theorem determines the region of achievable rates for efficient estimation of BSS.

Now consider a source with a convex mutual information function with respect to the unknown parameter θ. Figure 2.2 shows the mutual information function of such a source. The mutual information is a convex function of parameter θ with minimum at θ_m. The following theorem shows that the region of "achievable" rates for such sources is determined by the region of rates corresponding to the fixed parameter θ_m. Figure 2.3 illustrates the region of rates for two cases where $\theta = \theta_m$ and $\theta_s < \theta_m$ for this source. The area of the region is maximum for θ_m and shrinks with decreasing θ.

We now state the main theorem.

Theorem 2.7.2 *For any source $Q(X,Y;\theta)$ with marginal entropies independent of the parameters θ, if $I(X;Y,\theta)$ is a convex function of θ with a minimum at θ_m, the following rates are achievable (sufficient), i.e. the maximum attainable accuracies of distributed estimation and local estimation are equivalent:*

$$\begin{aligned} R_x &\geq H(X|Y;\theta_m), \\ R_y &\geq H(Y|X;\theta_m), \\ R_x + R_y &\geq H(X,Y;\theta_m), \end{aligned}$$

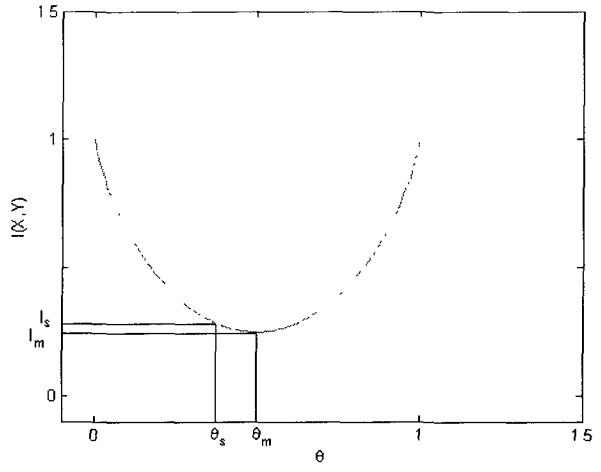


Figure 2.2: A convex mutual information function with minimum at θ_m. The mutual information for any parameter $\theta_s \leq \theta_m$ is more than I_m corresponding to θ_m.

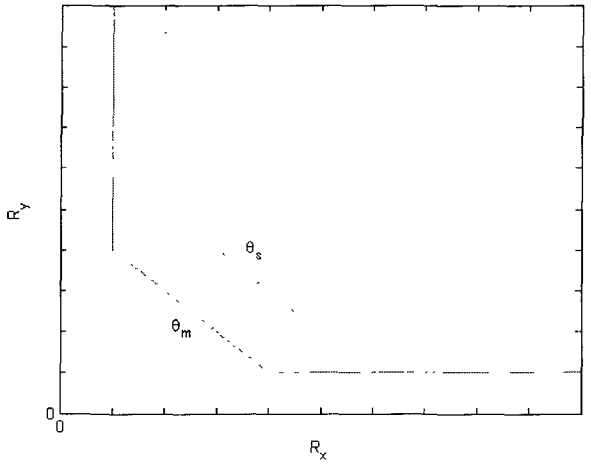


Figure 2.3: Region of rates corresponding to parameters θ_m and θ_s in Figure 2.2

where the entropies are with respect to the PD $Q(X,Y;\theta_m)$.

Proof 2.7.1 *Suppose for any* $\epsilon > 0$ *we fix* $\theta_s = \theta_m \pm \epsilon$. *Since* I *is a convex function of* θ *with minimum at* θ_m, *thus* $I(X;Y,\theta_s) \geq I(X;Y,\theta_m)$. *The rest of the proof is similar to the proof of Theorem 2.5.1 up to inequality (2.22). Now notice that if we substitute* $I(X;Y,\theta)$ *with* $I(X;Y,\theta_m)$ *in this inequality, since* $I(X;Y,\theta_s) \geq I(X;Y,\theta_m)$ *and the marginal entropies H(X) and H(Y) are not functions of* θ, *the condition of achieveability is* $R_x + R_y \geq H(X,Y;\theta_m) \geq H(X,Y;\theta_s)$ *which is always true and therefore the probability of error* $P(E_1)$ *(defined in (2.19)) decreases to zero exponentially surely. Similarly for* $P(E_2)$ *and* $P(E_3)$ *(defined in (2.20) and (2.21), respectively).*

Corollary 2.7.3 *For the BSS*(θ) *with* $0 \leq \theta \leq \theta_m$, *and* $0 \leq \theta_m < \frac{1}{2}$, *when side information is available* ($R_y \geq H(Y) = 1$), *the achievable region is:*

$$R_x > H_b(\theta_m),$$

where $H_b(\theta) = -(\theta \log \theta + (1-\theta)\log(1-\theta))$.

Proof 2.7.2 *Observe that for the BSS* $I(X;Y,\theta) = 1 - H(\theta)$ *which is a monotically decreasing convex function of* θ *for* $0 \leq \theta \leq \frac{1}{2}$. *Therefore by setting each value of* $\theta = \theta_m$ *(see Figure 2.4), we have* $H(X|Y) = H_b(\theta_m)$ *and therefore using the above theorem the region of rates is* $R_x \geq H_b(\theta_m)$.

Remark 12 *This result is similar to Amari's result presented in [88]. Since the region of rates is a convex set, once the theorem is applied for the corner points, the other points of the region can be treated similarly to determine a lower bound region for achievable rates. We presented their results here for completeness.*

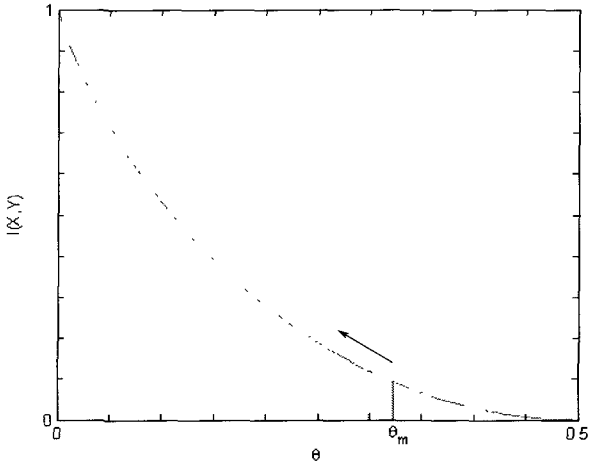


Figure 2.4: The mutual information function of BSS

2.8 Discussion

Distributed coding theorem of this chapter was the first step for determination of the rates under which a distributed estimator *can achieve* the same accuracy as it can in *local estimation*. We showed that for any fixed parameter, the Slepian–Wolf rates are not only *sufficient* [48] but also *necessary* for efficient estimation.

In this section, we first discuss the most important reference on distributed estimation literature to contrast the contribution of our distributed coding theorem. We show that the method of Han and Amari gives a set of *sufficient* rates for efficient estimation in distributed estimation (for fixed set of parameters). The theorem proved in this chapter, however, proved these sets are also *necessary*.

In the second note, we study a famous *"Modulo-two adder" source network* due to [61]. We show that in such a network, if communications is for the purpose of estimation, in contrast to the Slepian–Wolf rates, the triple set of $(0, 0, 0)$ rates are achievable.

2.8.1 A note on the method of Han and Amari [47]:

The most recent results in the literature on achievable rates in multiterminal distribution (more precisely two–terminal distribution estimation) are due to Han and Amari [48] in which the authors reviewed all previous literature on the subject[5]. Additionally, they completed the theory of multiterminal distribution for discrete sources in that paper- in the sense that they proved that provided a set of single-letterized rate-compatibility conditions are satisfied, asymptotically consistent and efficient distributed estimation of parameters is possible. The basic idea of the HA method is to find a set of rate–compatibility rates for each encoder that when satisfied, the transmission of sufficient information to the encoder is guaranteed to reconstruct the joint type (i.e. sufficient statistic) at the decoder. Based on these ideas, the authors prescribed a maximum likelihood estimation equation for estimation of parameters using the transmitted joint-type. They then computed the approximate distribution of the solution to the ML estimation equation and derived the Fisher information achievable when the rate-compatibility conditions are satisfied.

We begin with the following definitions.

Test channels and auxiliary variables: It is common practice in information theory to use auxiliary random variables defined as the output of an arbitrary stochastic mapping function, also known as a *test channel*. Test channels have the original random variables as their inputs. In two–terminal distributed estimation, two auxiliary random variables U_θ and V_θ are needed. These RV's are the output of the corresponding stochastic mappings, also referred to as *test channels*:

$$\begin{aligned} p_\theta(U|X) &: \mathcal{X} \to \mathcal{U} \\ p_\theta(V|Y) &: \mathcal{Y} \to \mathcal{V}. \end{aligned}$$

[5]To our knowledge, the only authors who referred to the Han and Amari works were Jornsten and Yu [55]. In their results, an approximate method is provided to make the HA method computationally accessible for special cases of Gaussian sources. We consider these results irrelevant to our purpose.

These variables are defined such that for all θ the Markov-chain property holds:

$$U_\theta \leftrightarrow X_\theta \leftrightarrow Y_\theta \leftrightarrow V_\theta,$$

It is important to notice that the auxiliary variables defined are such that they depend on θ only through the marginal distribution $Q(X;\theta)$ and $Q(Y;\theta)$. This is a crucial assumption which holds by construction and guarantees that the auxiliary variables are generated only by the knowledge of the marginal distributions, a constraint that discriminates distributed estimation from local estimation. The outputs of the test channels, i.e. (u^n, v^n) are usually called *codewords*. It is assumed that at least some of the information is preserved in such stochastic mappings, i.e.:

$$I(U_\theta; X_\theta) > 0, \quad I(V_\theta; Y_\theta) > 0.$$

Also, there is a standard assumption called the *positivity condition* [48], [55]:

$$Q(X, Y; \theta) > 0, \quad p_\theta(U, X, V, Y) > 0, \quad \forall u, x, y, v, \theta > 0.$$

Remark 13 *Note that all random variables and probability distributions are functions of the* unknown *parameter. This constraint is the main difficulty in the theory of multiterminal distribution. More specifically, only a* universal coding *scheme is desirable as if it is not universal, the coding scheme depends on an unknown parameter. The interesting result of Han and Amari is invaluable for it is asymptotically universal (independent of the unknown parameter θ and valid for any estimator including consistent estimators).*

When θ is known, it is obvious that any lossless distributed coding scheme can be used for multiterminal distribution. More specifically, as it is also noted in ([55], page 15), when θ is known at the encoders, any rate in the Slepian-Wolf region is

achievable:

$$\begin{aligned} R_x &\geq H_\theta(X|Y), \\ R_y &\geq H_\theta(Y|X), \\ R_x + R_y &\geq H_\theta(X;Y). \end{aligned} \tag{2.24}$$

However, when θ is not known at the decoder and is desired to be estimated, the intriguing question is whether these rates are also necessary. We showed that for any fixed value of the parameter, the answer to this question was also positive (see Theorem 2.5.1).

We review the HA method by reviewing the main theorem in [47] borrowed from [55], as the following.

Theorem 2.8.1 *Consider parameters $\theta \in \Theta$ where Θ is a compact subset of $\mathcal{R}^k$. Provided that the* rate-compatibility conditions*:*

$$\begin{aligned} R_x &\geq I(U_\theta; X_\theta|V_\theta), \\ R_y &\geq I(V_\theta; Y_\theta|U_\theta), \\ R_x + R_y &\geq I(U_\theta, V_\theta; X_\theta, Y_\theta), \end{aligned} \tag{2.25}$$

are satisfied, then there exist universal coding functions $f : \mathcal{U}^n \to \mathcal{X}^n$ and $g : \mathcal{V}^n \to \mathcal{Y}^n$, with rates $R_x = \lim_{n\to\infty} \frac{\log|f|}{n}$ and $R_y = \lim_{n\to\infty} \frac{\log|g|}{n}$ that achieve the maximum likelihood estimate $\hat{\theta}_{ML}$ with the following covariance:

$$C_{HA}(\theta|R_x, R_y) = J_{HA}^{-1} \tag{2.26}$$

$$J_{HA} = (\dot{\nabla} Q_\theta)_{M'} (H_{M'} G_{M'} H_{M'}^T)^{-1} (\dot{\nabla} Q_\theta)_{M'}^T. \tag{2.27}$$

where $Q_\theta = Q(X, Y; \theta)$. Here $\dot{\nabla} Q_\theta$ is partial derivative $p_\theta(u|x) \nabla Q(x, y; \theta) p_\theta(v|x)$, M is the index of observable types and the set M' is the subset of observable types that

are unconstrained *with respect to* M. *The matrix* $H_{M'}$ *is the projection matrix onto the unconstrained observable types. The matrix* $G_{M'}$ *is the unconstrained covariance matrix of the multinomial distribution* $p_\theta(U, X, Y, V)$.

Proof 2.8.1 *Refer to [47].*

Remark 14 *From a practical point of view, the rate–satisfiability conditions in the HA method are difficult to establish. They require a critical selection of proper test channels that are functions of the unknown parameter.*

Moreover, the construction of the likelihood equations is extremely difficult. The key element of the HA method is the projection matrix $H_{M'}$. *This operator characterizes the amount of information carried from* (x^n, y^n) *into the auxiliary random variables* (u^n, v^n) *under the coding scheme. The mathematical complexity required to compute this matrix is extremely high and a non–trivial exercise even for the simplest examples, e.g. binary symmetric source as noted by the same authors [47] and others ([55], page 75).*

Remark 15 *Suppose we set the test channels to the identity function. This is equivalent to setting* $U_\theta = X_\theta$ *and* $V_\theta = Y_\theta$. *We also fix* θ. *The rate-compatibility conditions of the HA theorem is in this case:*

$$\begin{aligned} R_x &\geq H_\theta(X|Y), \\ R_y &\geq H_\theta(Y|X), \\ R_x + R_y &\geq H_\theta(X, Y). \end{aligned}$$

In this case, it is shown [47] that the Fisher information is asymptotically equal to the Fisher information achieved in local estimation:

$$J_{HA} = (\nabla Q_\theta)_M G_M^{-1} (\nabla Q_\theta)_M^T,$$

where in this case all types are observable ($M' = M$). The HA theorem proves the sufficiency (achievability) *of the rates in a region similar to the Slepian-Wolf [90] region. However, this theorem does not discuss the* necessary *part, i.e. whether there exists an asymptotically efficient estimator if the rates do not satisfy the rate-compatibility conditions or equivalently, for a fixed θ, if they do not satisfy the Slepian-Wolf region. The results of our theorem proves the converse conjuncture.*

2.8.2 A note on "Modulo-two adder" source network:

Here we study a famous *"Modulo-two adder" source network* [61]. We show that in such a network, if communications is for the purpose of estimation, the triple set of $(0, 0, 0)$ rates are achievable.

We begin with the following definition: **"Modulo-two adder" source network** (Korner and Marton [61]) Let (X^n, Y^n) be n *i.i.d* samples from a discrete memoryless source $Q(X, Y; \theta)$ with $X \in \mathcal{X}^n$, $Y \in \mathcal{Y}^n$. Also let Z^n be the sequence defined by $Z = X \oplus Y$, the binary addition of samples (X^n, Y^n), where $Z \in \mathcal{Z}^n$. Here $\mathcal{X} = \mathcal{Y} = \mathcal{Z} = \{0, 1\}$. The sequences are encoded separately by three encoder functions $f : \mathcal{X}^n \to \Pi_f$, $g : \mathcal{Y}^n \to \Pi_g$, and $e : \mathcal{Z}^n \to \Pi_e$ with code rates $R_x = n^{-1} \log ||f||$, $R_y = n^{-1} \log ||g||$, and $R_z = n^{-1} \log ||e||$, respectively. The range of encoders, Π_f, Π_g, and Π_e are arbitrary *binary* sets. For arbitrarily small $\epsilon > 0$, a decoder function $h : \Pi_f \times \Pi_g \times \Pi_e \to \mathcal{Z}^n$ reconstructs the sequence Z^n with the probability of error defined as $\Pr(h(f(X^n), g(Y^n), e(Z^n)) \neq Z^n) < \epsilon$. The rate triple (R_x, R_y, R_z) is called *achievable* if there exists at least one triple of encoders (f, g, e) and a decoder h with probability converging to 1, by which one can construct the sequences of codes that provide transmission of the sequence Z^n to the receiver with probability of error converging to $\epsilon \to 0$ as n becomes sufficiently large.

Lemma 2.8.2 *For the* binary symmetric source $Q(X, Y; \theta)$ *the following rates are*

achievable:

$$\begin{aligned} R_x + R_z &\geq H(Z), \\ R_y + R_z &\geq H(Z). \end{aligned}$$

Proof 2.8.2 *Refer to [61].*

This theorem provides the achievable rates when the transmission of the modulo-two sum of the sequences (X^n, Y^n) is desired.

We study a similar source network setting, for the case when the distributed estimation of the source parameter θ is desired. The discussion is for general binary sources, including the BSS. We first show that the *types* of the sequences X^n, Y^n and Z^n are sufficient statistics. Then we conclude that the triple rate $(R_x, R_y, R_z) = (0, 0, 0)$ is achievable (in the sense of distributed estimation of θ).

Lemma 2.8.3 *Given* n i.i.d. *samples* (X^n, Y^n) *taken from a binary PD* $Q(X, Y, \theta)$, *the marginal types* $\tilde{p}_{x^n}(1)$ *and* $\tilde{p}_{y^n}(1)$ *and the joint-type* $\tilde{p}_{x^n y^n}(1, 1)$ *are* minimal *sufficient statistics.*

Proof 2.8.3 *The bivariate binary source with PD* $Q(X, Y, \theta)$ *is a member of the* curved exponential family*:*

$$Q(X, Y; \alpha(\theta)) = \exp\left[C + \sum_{k=1}^{3} \alpha^k F_k(X, Y) - \psi(\alpha))\right], \tag{2.28}$$

where $(\alpha(\theta) \in \mathcal{R}^3)$ *is the vector of* natural or canonical parameters, *and the functions* $F_k(X, Y)$ *are called the* sufficient statistics. $\psi(\alpha)$ *is the normalization function:*

$$\psi(\alpha) = \log \int \exp\left[C(x, y) + \sum_{k=1}^{K} \alpha^k F_k(x, y)\right] dxdy. \tag{2.29}$$

Here $C(X, Y) \triangleq 0, F_1(X, Y) \triangleq \delta_1(x), F_2(X, Y) \triangleq \delta_1(y), F_3(X, Y) \triangleq \delta_{11}(x, y)$, *where* $\delta_a(x) = 1$ *iff* $(x = a)$, *is the* Kronecker delta *function and* $\delta_{11}(x, y)$ *is defined*

similarly (see 2.2). The canonical parameters, also referred to as α-coordinates, are defined as:

$$\alpha^1 \triangleq \log \frac{p_{10}}{p_{00}}, \tag{2.30}$$

$$\alpha^2 \triangleq \log \frac{p_{01}}{p_{00}}, \tag{2.31}$$

$$\alpha^3 \triangleq \log \frac{p_{00}p_{11}}{p_{01}p_{10}}, \tag{2.32}$$

and

$$\psi(\alpha) = -\log p_{00} = \log(1 + e^{\alpha_1} + e^{\alpha_2} + e^{\alpha_1}e^{\alpha_2}e^{\alpha_3}) \tag{2.33}$$

It can be shown by verification that α^1, α^2, and α^3 are affinely independent. Also by definitions F_1, F_2, and F_3 are affinely independent. Therefore [14] the exponential representation of $p(X, Y; \theta)$ is minimal, the α-parameters are "minimal canonical" *parameters and functions F_1, F_2 and F_3 are the* minimal sufficient statistics.

Remark 16 *Note that F_1, F_2, and F_3 are in fact the relative occurrences of $(x = 1)$, $(y = 1)$ and $(x, y) = (1, 1)$, respectively. This shows that the relative frequency of occurrence of "1" in the sequences x^n and y^n as well as the relative frequency of joint occurrences of "$(1, 1)$" in (x^n, y^n) are sufficient statistics for estimation of the PD parameters.*

Corollary 2.8.4 *When the source is symmetric, i.e. $Q(X, Y; \theta) = BSS(\theta)$, the joint-type $\tilde{p}_{x^n y^n}(1, 1)$ is the sole sufficient statistic.*

Proof 2.8.4 *For the binary symmetric source, the marginal distributions are known at the decoder, i.e. $p_{x1} = p_{y1} = \frac{1}{2}$. Therefore, according to lemma (2.8.3) the joint-type p_{11} is the sole minimal sufficient statistic.*

Lemma 2.8.5 *Given n* i.i.d. *samples (x^n, y^n) taken* i.i.d *from a binary source $Q(X, Y; \theta)$, and a sequence Z^n defined as the modulo-two sum of (X^n, Y^n), the marginal types p_{x1}, p_{y1} as well as the type p_{z1} are the* minimal *sufficient statistics.*

Proof 2.8.5 *It is easy to verify that* $p_{11} = \frac{1/2}{(} p_{x1} + p_{y1} - p_{z1})$. *Therefore* p_{x1}, p_{y1} *and* p_{z1} *provide the set of sufficient statistics of Theorem 2.8.3.*

Theorem 2.8.6 *For the modulo-two sum source network, and for* efficient *distributed estimation of the source* $p(X, Y; \theta)$, *the triple rate* $(R_x, R_y, R_z) = (0, 0, 0)$ *is achievable.*

Proof 2.8.6 *It is a well-known fact since the cardinality of types is polynomial in* n, *the types can be transmitted with arbitrarily small error by* zero *rate. Therefore the triple of rates for transmission of marginals* p_{x1}, p_{y1}, *and* p_{z0} *are achievable.*

Remark 17 *It is important to notice that the "Modulo-two sum" source network of Korner and Marton is a special case of the* two-help-one *source network [61]. This network has been considered an instance of distributed coding scenarios (For example refer to [44], page 43). However, the existence of the modulo-two sum sequence represents a collaboration between the two parties* X *and* Y. *This collaboration lies in the fact that the random variable* Z *carries the* joint-type *of the the* (X, Y) *sequence pairs. This is the reason why the rates achievable in this case do not obey the results of the Slepian-Wolf [90] theorem. Also, when the two parties* X *and* Y *do not help* Z, *the achievable rate for transmission of* Z *is an obvious result of Shannon's source coding theorem* $(R_z \geq H(Z))$

Chapter 3

Distributed Parameter Estimation

3.1 Introduction

In this chapter, a low complexity algorithm for distributed maximum likelihood estimation of a binary symmetric source (BSS) using side–information is proposed. The estimation is formulated as an *incomplete–data problem* and is solved by the expectation–maximization (EM) algorithm. A low–complexity implementation of the algorithm using *coset codes* and LDPC-based *syndrome decoding* with message passing over a factor–graph is also proposed. The algorithm is a generalization of the LDPC-based syndrome decoding algorithm for the case when the probability distribution of the source is not known *a-priori*. Hence, the algorithm may be considered as a tool for achieving the corner points of the Slepian-Wolf (SW) region in distributed coding when the correlation channel information is not available.

Suppose two sequences (x^n, y^n) drawn $i.i.d$ from a binary symmetric source ($BSS(\rho)$), $p(X, Y; \rho)$, are encoded separately with code rates (R_x, R_y) and transmitted to a common decoder. The probability distribution (PD) of source is parameterized by the scaler ($\rho \in \mathcal{R}$). It is assumed throughout this chapter that $R_y \geq H(Y)$, and hence

the side–information sequence y^n is available at the decoder perfectly. It is also assumed that x^n is encoded (compressed) to a sequence u^m, where $R_x = m/n \leq 1$. It is assumed that the channel codes are chosen properly such that the communication channel can be considered to be noiseless, i.e. the focus is on the coding/decoding scheme for distributed source coding.

In distributed *source coding* with side–information (SI) at the decoder, the main objective is to reconstruct the sequence x^n using the SI y^n and *perfect* knowledge of the correlation between two random variables– the underlying PD or equivalently ρ. The Slepian-Wolf (SW) theorem [90], in this case, determines the achievable rate ($i.e. R_x \geq H(X|Y)$) for near-lossless reconstruction of the sequence x^n. Practical code design for distributed coding with SI was initiated when Wyner [102] recognized the similarity of the problem with channel coding. He observed that the SI y^n may be considered as output of a hypothetical *correlation channel* with input x^n. Thus, he predicted that the channel coding techniques may be used for distributed coding using SI. The *distributed coding with syndromes* (DISCUS) [85] [78] was the first practical code that implemented Wyner's idea. It was shown that this approach not only can approach the rates of the corner points of the SW region, but in general all SW rates [86]. The low–complexity extension of DISCUS, e.g. LDPC-based codes [66], distributed Turbo codes [12][74] was also shown to be capable of achieving all points in the SW region. For a review on distributed source coding refer to ([104] and the references therein).

In distributed *estimation* using SI, on the other hand, the objective is to estimate the source parameter ρ at the decoder using a *compressed* version of x^n and the SI y^n. The difficulty here is due to the fact that x^n needs to be encoded and decoded without knowledge of its correlation with the SI (i.e. the source parameters ρ). More specifically, the encoding process relies upon the marginal distribution of the source, $p(X)$, which is in general dependent of the (*unknown*) parameter ρ. Therefore, the

decoding must not depend on the unknown correlation information ρ. Proposing a *universal coding* scheme for distributed estimation is in general a challenging problem [48]. Here we concentrate on design conditions for which the distributed estimator that can achieve the same accuracy that can be achieved in local estimation– when sufficient statistics (uncompressed sequences) are available completely at the estimator. For the BSS, a theorem due to Amari [88], studied in the following, provides guidelines to achieve this goal.

As stated before, distributed estimation of $BSS(\rho)$ using SI may be considered as a distributed source coding problem using SI when the source parameters are not known *a priori*. This analogy is used here to extend the DISCUS [79] algorithm used in distributed source coding to the case when the statistics between the sequence x^n and the SI y^n are not known at the decoder. Suppose the sequence x^n is compressed to u^m where $m \leq n$. The main idea is to treat the available data (y^n, u^m) as *incomplete–data* and the compressed sequence x^n as the *hidden* sequence. In this chapter, the distributed estimation scenario is formulated as am ML estimation problem using incomplete–data, and the expectation–maximization (EM) algorithm [35] is used to provide an iterative estimate of the source parameter ρ.

The LDPC–based low–complexity extension of this idea for long–sequences (which are inevitable for efficient estimation) is also presented. A factor–graph for LDPC–based syndrome decoding algorithm is equipped with an extension that implements the EM algorithm. In the augmented graph the posterior probabilities computed in syndrome decoding are used in the E-step to implement the expectation operations needed in the E–step. The following M–step estimates a new value for the unknown parameter.

More specifically, assuming that SI y^n is available at the decoder, the source is encoded using a *coset code* characterized by an LDPC parity–check matrix. *Syndrome decoding* is implemented by message passing over a factor–graph to compute

the posterior distribution of encoded messages, given the SI y^n and the received syndrome u^m. The factor–graph is equipped with an extension that implements the EM–algorithm. This incorporates the computed posterior probabilities in an iterative fashion for estimating the unknown source parameter.

In addition to distributed estimation, the proposed algorithm may be used for distributed source coding. The success of all distributed source coding schemes relies upon knowledge of the correlation between the sequence pairs. These algorithms do not perform optimally when this correlation information is not available at the decoder. The proposed algorithm may be used for joint correlation parameter estimation and distributed syndrome decoding. In this case, the algorithm may be considered as an example of a *universal distributed coding/decoding* scheme.

In Section 3.2, the problem statement is presented and the expectation–maximization (EM) algorithm is formulated to solve the maximum likelihood (ML) estimation of the source parameter using SI. In Section 3.3, fundamental concepts and definitions in linear block parity check codes are reviewed and in Section 3.4 their application for distributed source coding is studied. In Section 3.5 Wyner's idea for distributed source coding and the DISCUS algorithm are extended to distributed estimation using the EM algorithm. Moreover, since long sequences are required for efficient estimation, the proposed algorithm needs to be implemented using low–complexity codes. For this purpose, a short tutorial on LDPC codes and message passing over factor–graphs is presented in Section 3.6. Then in Section 3.7 the basic algorithm proposed in Section 3.5 is extended to LDPC–based source coding using syndrome decoding by message passing over factor graphs. It is shown in this section that the LDPC decoding scheme needs to be modified properly when used for syndrome–decoding. This is studied in Section 3.7.1. The estimation efficiency is studied by comparing the estimation variance with the achievable Fisher information in multiterminal estimation theory. The achievable rates are studied in Section 3.8. Simulation results

for distributed estimation of sources are presented in Section 3.9. It is also shown through simulations that the proposed algorithm may be used as a tool for achieving the corner points of the Slepian-Wolf (SW) region [90] for joint–estimation and decoding in distributed coding.

Notation: All the binary vectors in this chapter are assumed to be row-vectors. Also all logarithms are arbitrarily chosen to be base 2. A random variable and its particular realization are represented by uppercase and lowercase letters, X and x, respectively. A sequence (vector) of n random variables and their any particular realizations are shown by superscript n, e.g. X^n and x^n, respectively. Proper subscripts are used, e.g. X_i and x_i, to reference any element i of a random sequence and its realization, respectively. When a particular n-sequence is needed to be referenced, where clear from context, the superscript n is replaced by the reference index, e.g. $x^j \triangleq x^{nj}$.

3.2 Distributed ML Estimation with Side Information

We begin with a definition.

3.2.1 Binary symmetric source

Suppose $\mathcal{X}$ and $\mathcal{Y}$ are sets with finite or countably infinite cardinality. A binary symmetric source (BSS) with parameter ρ, $(BSS(\rho))$, is a double binary source with joint probability distribution over a pair of random variables $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ defined as the following:

$$p(X, Y; \rho) = \begin{pmatrix} \frac{\rho}{2} & \frac{1-\rho}{2} \\ \frac{1-\rho}{2} & \frac{\rho}{2} \end{pmatrix}, \tag{3.1}$$

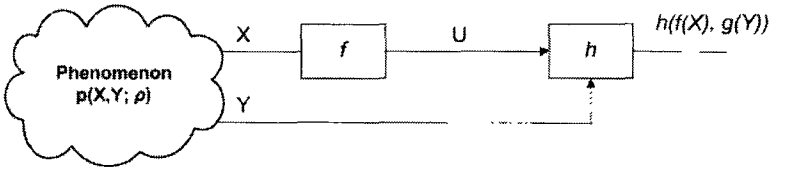


Figure 3.1: Distributed estimation with side–information

where without loss of generality we assume $0 < \rho \leq \frac{1}{2}$. For the $BSS(\rho)$ we have:

$$\begin{aligned} H(X|Y) = H(Y|X) &= H_b(1-\rho) \\ I(X;Y) &= 1 - H_b(1-\rho) \\ H(X) = H(Y) &= 1, \end{aligned}$$

where $H(X)$ is the entropy of the random variable X and $I(X;Y)$ is the mutual information between two random variables X and Y. Also $H_b(\rho) \triangleq -\rho \log \rho - (1-\rho)\log(1-\rho)$.

3.2.2 Problem Statement

Let n samples drawn $i.i.d$ from a $BSS(\rho)$ are encoded independently at rates R_x and R_y, respectively, and transmitted to a common decoder. We assume $R_y \geq H(Y)$ and hence the sequence y^n, referred to as *side–information* (SI), is available with arbitrarily small error at the decoder. The sequence x^n is encoded into message $u^m = f(x^n)$ $(m \leq n)$ with rate restriction ($R_x = \frac{m}{n} \leq R_c$ where R_c is the available capacity of the channel). Given a limited rate $0 \leq R_c \leq H(X)$ for transmission of x^n, it is desired to design a pair of low–complexity encoder–decoders to optimally estimate the source parameter ρ (see Figure 3.1).

For unbiased (asymptotically consistent) estimation, the mean square error is

equal to the variance of estimation defined as:

$$V_n(\rho, R_x, R_y) = E_\rho[(\hat{\rho} - \rho)^2] \geq J^{-1}(\rho, R_x, R_y).$$

where E_ρ is *expectation* with respect to $p(X, Y; \rho)$. Here $J(\rho, R_x, R_y)$ is the *Fisher information* (FI) with respect to the parameter ρ. The inverse of the FI defines a lower bound on the attainable variance in estimation (read "accuracy"), also referred to as the Cramer-Rao lower bound (CRLB). In distributed estimation, the CRLB is not only a function of the parameters, but a function of the available rates.

In what follows, we are mainly interested in scenarios where distributed estimation is equivalent to *local* estimation. "Equivalent" is in the sense that, given the complete SI y^n, the accuracy of estimation using the compressed sequence u^m is as good as the accuracy when the uncompressed sequence x^n is used.

3.2.3 Binary symmetric channel model and MLE

In order to define the likelihood function, we use Wyner's *correlation channel* between two correlated random variables [102]. We consider Y as the output of a hypothetical binary symmetric channel (BSC) with input X and noise W:

$$Y = X \oplus W. \tag{3.2}$$

where $\oplus$ indicates the *modulo-two* summation. Here, W is a *Bernouli*(ρ) process, i.e. a binary random variable with $\Pr(W = 1) = \rho$. The parameter ρ is usually called *the correlation parameter.*

Given an input sequence x^n of n i.i.d samples to the BSC, the likelihood of the output sequence y^n is:

$$\begin{aligned} p(y^n|x^n;\rho) &= \rho^{d_H(x^n,y^n)}(1-\rho)^{n-d(x^n,y^n)} \\ &= \prod_{i}^{n} \rho^{z_i}(1-\rho)^{(1-z_i)}, \end{aligned} \tag{3.3}$$

where $d_H(.)$ is the *Hamming distance*:

$$d_H(x^n, y^n) = \sum_{i=1}^{n} (x_i \oplus y_i).$$

We also define the binary random variable $Z = Y \oplus X$.

Remark 1 *With the BSC interpretation, BSS parameter estimation is in fact a means for estimating the correlation between two sequences. Therefore, throughout, the accuracy of BSS parameter estimation is also an indication of the success of estimation of this correlation.*

3.2.3.1 MLE with complete data

When complete knowledge of the pair (y^n, x^n) is available, the log–likelihood is:

$$\begin{aligned} \mathcal{L}(\rho) &= \log p(y^n | x^n; \rho) \\ &= \sum_{i=1}^{n} \Big[z_i \log \rho + (1 - z_i) \log(1 - \rho) \Big]. \end{aligned} \tag{3.4}$$

Thus, the ML estimate is obtained as:

$$\hat{\rho}_{ML} = \frac{\sum_i^n z_i}{n}. \tag{3.5}$$

The mean and the variance of estimation are:

$$\mathrm{E}\{\hat{\rho}_{ML}\} = \frac{\sum_i^n \mathrm{E}(z_i)}{n} = \frac{n\rho}{n} = \rho,$$

$$\mathrm{V}\{\hat{\rho}_{ML}\} = \frac{\sum_i^n \mathrm{V}(z_i)}{n^2} = \frac{\rho(1-\rho)}{n}.$$

In this case, the estimator achieves the CRLB defined as the inverse of the FI:

$$J = E\Big[\frac{\partial^2 \mathcal{L}(\rho)}{\partial \rho^2}\Big] = \frac{n}{\rho(1-\rho)}.$$

3.2.3.2 The MLE with compressed data using side–information

Ideally, complete knowledge of sufficient statistics, (y^n, x^n), is necessary for estimation of ρ. However, in distributed estimation with SI, the available data at the decoder consists only of the SI y^n and the compressed sequence u^m. Therefore, ML distributed estimation using SI can be considered as an instance of the so–called *incomplete–data problem.* The exact solution of the MLE with incomplete data (y^n, u^m) is NP hard. Given a compressed sequence u^m, the number of candidate sequences x^n that could have been compressed into u^m increases exponentially with n. Therefore, any exhaustive search in the input space has an exponentially increasing complexity.

In contrast, the EM algorithm [35] provides a computationally efficient method for solving the problem approximately. More specifically, instead of maximizing the complete–data likelihood, the EM algorithm maximizes the expectation of the likelihood given the partially available data. For this purpose, the EM algorithm alternates between two main steps. In the expectation (E) step, the algorithm computes a lower–bound on the likelihood of available data. Then, in the maximization (M) step, this lower–bound is maximized to solve for a new value of the parameter.

The log–likelihood of the available data (y^n, u^m) can be lower–bounded in the following way:

$$\begin{aligned} \mathcal{L}(\rho) &= \log p(y^n|u^m;\rho) \qquad (3.6) \\ &= \log \sum_{x^n} p(y^n, x^n|u^m;\rho) \\ &= \log \sum_{x^n} p(y^n|x^n, u^m;\rho) p(x^n|u^m;\rho) \\ &= \log \sum_{x^n} p(y^n|x^n;\rho) p(x^n|u^m;\rho) \qquad (3.7) \end{aligned}$$

We assume a uniform sampling over the members of each coset and therefore substitute ($p(x^n|u^m;\rho) = K_1$). We have:

$$\begin{aligned}
\mathcal{L}(\rho) &= \log \sum_{x^n \cdot u^m = f(x^n)} K_1 p(y^n|x^n;\rho) \\
&= \log \sum_{x^n : u^m = f(x^n)} K_1 \frac{p(x^n)}{p(x^n)} p(y^n|x^n;\rho) \\
&= \log \sum_{x^n : u^m = f(x^n)} K_1 \frac{p(y^n, x^n;\rho)}{p(x^n)} \\
&\geq \sum_{x^n : u^m = f(x^n)} \log p(y^n, x^n;\rho) - \sum_{x^n : u^m = f(x^n)} \log p(x^n) + K_2, \qquad (3.8)
\end{aligned}$$

where we used the *Jensen's inequality* in the last equation. We defined the constant $K_2 = \sum_{x^n : u^m = f(x^n)} \log K_1$. Note that the second and third terms in (3.8) do not play any role in the MLE.

At iteration t, the parameter is assumed to be known and equal to ρ_t. In the E–step, the expectation of the log–likelihood, represented by the $\mathcal{F}$ function defined below is computed. The expectation is with respect to the posterior distribution of the input given the available data, and is computed using the current value of the parameter, ρ_t:

$$\textit{E-step:} \quad \mathcal{F}(\rho|\rho_t) = \sum_{x^n : u^m = f(x^n)} p(x^n|y^n, \rho_t) \log p(y^n, x^n;\rho) \qquad (3.9)$$

Note that the $\mathcal{F}$ function is the expectation of the first part of the lower bound in Equation 3.8, the only part that is a function of the parameter ρ.

In the M–step the $\mathcal{F}$ function is maximized to obtain a new value for the parameter:

$$\textit{M-step:} \quad \hat{\rho}_{(t+1)} = \arg\max_{\rho} \mathcal{F}(\rho|\rho_t). \qquad (3.10)$$

In words, in the E–step, given the initial value of the unknown parameter, the posterior distribution of the most–likely input sequences x^n responsible for generating

the compressed sequence u^m is computed. This distribution is used to compute the expectation of the log–likelihood function. Then in the M–step, this expectation is maximized to re–estimate a new value of the unknown parameter. The new value of the parameter is used in the next E–step. The algorithm continues until convergence. With a proper initial point, the EM algorithm guarantees an increase (or at least no reduction) in likelihood in each iteration.

In what follows, we implement the EM algorithm for distributed estimation with SI using coset–code linear block codes with syndrome decoding currently used in distributed source coding. For this purpose, we first review fundamental concepts in coding. This begins with *linear block parity check* codes for which we discuss maximum likelihood (ML) detection, syndrome decoding and the notion of *coset(s)*. A quick review of the application of linear block codes for distributed coding follows. When the necessary background is laid out, the application of the linear block parity check codes for distributed estimation of source parameters using the EM algorithm is presented. This algorithm may be considered an extension of previously proposed distributed coding algorithms (more specifically DISCUS [79]) to the case when the correlation information is not available *a priori*.

3.3 Linear Block Parity Check Codes

In general, a linear block parity check (n, k) code transforms k bits of input sequences, represented by vector b^k, into an $n-$vector $(n \geq k)$ codeword though a linear operation defined by a generator matrix $G \in \{0,1\}^{k \times n}$. There are $(n-k)$ parity check bits added to the information bits to protect them from the noise present in transmission. The encoded information bits are then recovered using a linear operation defined by a parity check matrix $H \in \{0,1\}^{(n-k) \times n}$. The coding and decoding in LBPC codes can be explained better with a familiar example, the Hamming codes [51], as follows.

3.3.1 Example– Hamming $(7,4)$ channel code

The *Hamming code* is a linear parity check code with $n = 2^m - 1$ with $m \geq 2$. For this code m is the number of parity check bits and $k = 2^m - 1 - m$. The minimum distance for this code is $d_{min} = 3$. The columns of H consists of all *non-zero* vectors of length m. For the Hamming $(7,4)$ code it is desired to encode $k = 4$ bits of information in b^k into a $n = 7$ bit codeword vectors with $m = n - k = 3$ parity check bits. Therefore:

$$G = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \quad H = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

The codewords generated by the generator matrix:

$$x^n = b^k G$$

are in a linear subspace spanned by the rows of G. There are $C = 2^k = 2^4$ binary permutations of $k = 4$ bits in the information vector b^k and thus the same number of codewords. Since by definition $GH^T = 0$, for any codeword x^n:

$$x^k H^T = 0,$$

i.e. all codewords are orthogonal to the rows of H and thus are in the *null space* of H. The space of possible vectors is an $(n = 7)$ dimensional binary lattice. The generator matrix assigns to each input information vector a vertex of this lattice, and so chooses $C = 2^k = 16$ vertices of this lattice as codewords. The most important characteristic of the Hamming code is that G performs this mapping onto a linear subspace consisting of vertices with minimum *Hamming distance* greater than $d_{min} = 3$. In other words, all $C = 16$ codewords of Hamming code are on vertices with minimum distance $d_H = 3$. This property, when accompanied with proper maximum likelihood detection, gives rise to the error-correction capability of the Hamming code (and in general linear block codes).

3.3.2 Maximum likelihood detection

The decoding capability of the Hamming code can be shown by an example. Suppose single bits x and y are the input and output of a binary symmetric channel $BSC(\rho)$, respectively. As we showed in Section 3.2.3, this is equivalent to writing $y = x \oplus w$, where w is a $Bernouli(\rho)$ random noise. For a vector of n $i.i.d$ bits x^n input to the BSC(ρ), the output vector y^n is $y^n = x^n \oplus w^n$, and the likelihood of the received sequence y^n is a binomial distribution:

$$\mathcal{L}(y^n, \rho) = p(y^n|x^n) = \rho^{d_H(x^n,y^n)}(1-\rho)^{n-d_H(x^n,y^n)},$$

where as before $d_H(x^n, y^n)$ is the Hamming distance between the two sequences. It is easy to verify that for $0 \leq \rho \leq \frac{1}{2}$ the likelihood is a decreasing function of d_H and is maximized for an input codeword with the minimum distance to y^n.

3.3.3 Syndrome and coset

Despite its apparent simplicity, the ML detection is an NP-hard problem. For large sequences, detection involves a search through the n-dimensional binary lattice that tends to become exponentially complex. A sub–optimal solution is the so-called *syndrome decoding*. The main idea is based upon the fact that all valid codewords lie in the *null–space* of H. The *syndrome* of a received vector is a measure of the violation of the vector from this null–space and is defined as:

$$\begin{aligned} s &= y^n H^T \\ &= (x^n + w^n)H^T \\ &= 0 + \mathrm{w^n H^T} \\ &= w^n H^T. \end{aligned}$$

The following two important properties [51] are needed.

Lemma 3.3.1 *The syndrome depends only on the error pattern and not the transmitted codeword.*

Proof 3.3.1 *The proof is immediate by observing that* $s^{(n-k)} = y^n H^T = w^n H^T$.

Lemma 3.3.2 *All error patterns that differ by a codeword have the same syndrome.*

Proof 3.3.2 *Let* $W = \{e^j : e^j = w^n + x^j, \quad j = 0, 1, ..., 2^k - 1\}$ *be the set of error vectors formed by adding* 2^k *codewords to a particular error vector* w^n. *We have:*

$$\begin{aligned} e^j H^T &= (w^n + x^j) H^T \\ &= w^n H^T. \end{aligned}$$

In words, associated with each syndrome is an error vector with the property that when added to any codeword, it will generate the same syndrome. This is the basis for the syndrome decoding scheme explained in the following. We first define cosets.

Let w^j be the error vector corresponding to syndrome s^j. We call the set of sequences formed by adding the error vector to all codewords a *coset*, with w^j as the *coset leader*. According to lemma 3.3.2 above, all the members of the coset have the same syndrome. There are 2^{n-k} syndromes and thus the same number of error patterns. As can be seen in the following table, the space of 2^n sequences are partitioned into 2^{n-k} sets (rows as cosets), each consisting of 2^k sequences (for clarity we omit the superscript n for vectors). Each coset (row) in this table consists of sequences with the same syndrome. In other words, each member of a coset when multiplied with the parity check matrix generates the same syndrome and is associated with the same error pattern. The first row associates with $w^n = 0$ syndrome and consists of channel codes words. We refer to this coset as the *original coset* in the

following discussion.

$$\begin{array}{ccccccc}
w^1 = 0 & x^2 & x^3 & \ldots & x^i & \ldots & x^{2^k} \\
w^2 & x^2 + w^2 & x^3 + w^2 & \ldots & x^i + w^2 & \ldots & x^{2^k} + w^2 \\
w^3 & x^2 + w^3 & x^3 + w^3 & \ldots & x^i + w^3 & \ldots & x^{2^k} + w^3 \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
w^j & x^2 + w^j & x^3 + w^j & \ldots & x^i + w^j & \ldots & x^{2^k} + w^j \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
w^{2^{n-k}} & x^2 + w^{2^{n-k}} & x^3 + w^{2^{n-k}} & \ldots & x^i + w^{2^{n-k}} & \ldots & x^{2^k} + w^{2^{n-k}}
\end{array}$$

It can be seen that the members of each coset are codewords offset by the same error vector. For any given channel the probability of a decoding error is minimized if the most likely error patterns are chosen as coset leaders [51]. In the case of the BSC, the error patterns with the smallest Hamming weight are most likely and therefore should be chosen as coset leaders.

For our example of the $(7, 4)$ Hamming code, the space of all sequences consists of $2^n = 2^7$ vectors. There are $2^{n-k} = 2^3$ cosets each having $2^k = 2^4$ vectors. The coset with leader $w^n = 0_{1\times 7}$ is the coset of error–free codewords, also referred to as the "*original coset*"

3.3.4 Syndrome decoding

Now, suppose a sequence y^n with syndrome $s^{(n-k)} = y^n H^T$ is received. The decoded sequence can be computed by:

$$\hat{x}^n = y^n \oplus \hat{w}^n, \tag{3.11}$$

where $\hat{w}^n$ is the coset leader of the coset corresponding to the received syndrome. The input information bits b^k can be extracted from the detection sequence $\hat{x^n}$.

Remark 2 *The decoding process in linear block codes tends to become complex as the code length increases. For ML detection, a lattice search through all codewords*

becomes practically impossible when the code length becomes very large. Additionally construction of a look–up table relating syndrome vectors to error patterns (that should be used to extract the input bits from the received vector) becomes exponentially complex with n. Low–density–parity–check (LDPC) codes were designed to provide a practical algorithm for decoding when the code length is very large. The complexity of LDPC decoding is linear with code–length [42]. We will study LDPC codes in following sections.

3.4 Linear Block Codes for Distributed Coding

The Slepian–Wolf [90] theorem provides the region of achievable rates for near-lossless reconstruction of two correlated sources. However, it does not provide a constructive solution to practical code design. In his seminal paper, Wyner [102] opened new directions in practical code design for this problem. He observed the similarity of decoding in the presence of SI with channel coding. Suppose we have two correlated random variables (X, Y) from the distribution $p(X, Y)$. By his analogy, the side–information Y can be considered as the output of a noisy channel with input X. Thus, he proposed using linear block channel codes for encoding and decoding of X using Y. He also showed that the corner points of the SW region, $(H(X|Y), H(Y))$ can be achieved by coset codes and syndrome decoding. In fact, for a received syndrome S, the decoder used Y together with the knowledge of the correlation between X and Y to determine which codeword in the designated coset was input to the "channel". So if the linear channel code was a good channel code for the *hypothetical* "correlation channel", then respective coset code was also a good source code for this type of correlation.

Perhaps due to the implementation complexity of this approach, these ideas were abandoned for decades until the recent growing interest in distributed processing

and sensor networks emerged. More specifically, it was Pradhan and Ramchandran that recently proposed a practical implementation of Wyner's idea for distributed coding [85]. The introduction of *distributed source coding using syndromes* (DISCUS) triggered a new wave of research on practical code design for distributed coding (For example see [12] and [74] for the application of Turbo codes for distributed coding and [66] for the LDPC-based codes for distributed coding). We illustrate the basic idea behind DISCUS by a simple example from [85]. Then we review some extensions to this example. [1]

3.4.1 Example: The Hamming $(7, 4)$ distributed source code

The Hamming channel code may also be used for distributed coding of correlated sources [85]. Here we represent the Hamming code by (n, k) with $m = n - k$ referring to the number of syndrome bits. Let x^n and y^n be two discrete memoryless uniformly distributed $n = 7$-bit binary random variables which are correlated in the sense that their Hamming distance is less than 1 bit, i.e. $d_H(x^n, y^n) \leq 1$. Observe that:

$$\begin{aligned} H(x^n) = H(y^n) &= 7 \text{ bits} \\ H(x^n|y^n) = H(y^n|x^n) &= 3 \text{ bits} \\ I(x^n; y^n) &= 4 \text{ bits} \\ H(x^n, y^n) &= 10 \text{ bits.} \end{aligned}$$

According to the SW theorem, x^n and y^n can be separately compressed at rates R_x and R_y, respectively, such that $R_x \geq 3$, $R_y \geq 3$, and $R_x + R_y \geq 10$ bits. Now let us consider the corner point of the rate region with $(R_x = H(x^n|y^n) = 3,\ R_y = H(y^n) = 7)$. For these rates, y^n can be considered fully available at the decoder. On the other hand, the

[1] It is important to notice that linear block codes are proved to be sufficient for achieving all points in SW region [30] provided that minimum entropy decoding is used. However, we must still investigate whether ML estimation performed with LDPC codes can achieve the same performance.

Hamming parity check matrix partitions the space of x^n vectors into $2^{n-k} = 2^3$ cosets (bins) each consisting of $2^k = 2^4$ codewords. In each coset, the distance between each pair of codewords is at least $d_H \geq 3$. Therefore, when y^n is known, there is one and only one codeword in each coset for which $d_H(y^n, x^n) \leq 1$, i.e. that matches the correlation channel model between X and Y. Therefore the rates of the corner point are achieved (other points of the region can be achieved either by time–sharing between the sources or symmetrical distributed source coding schemes proposed elsewhere e.g. see [45], [78], and [91]).

Let us illustrate the underlying concept with an example. Suppose x^n and y^n differ only in the 4^{th} bit:

$$\begin{aligned} x^n &= \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}, \\ y^n &= \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}. \end{aligned}$$

The syndrome for encoding x^n is:

$$s^m = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}^T = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}.$$

We defined $m = n - k$. For $(R_y = \frac{7}{7} = 1)$ the sequence y^n is completely known at the decoder. The syndrome s^m is transmitted to the decoder with a rate $R_x = \frac{3}{7}$. All codewords in the coset designated by s^m are at least $d_H = 3$ far apart. Therefore, having y^n accessible, as well as using the fact that x^n is not further than $d_H = 1$ from y^n guarantees that the true codeword x^n can be found in the coset.

The idea illustrated in this example is generalized for linear block parity check codes as follows.

3.4.2 Distributed source coding using syndrome (DISCUS)

Consider the case when x^n and y^n are equiprobable n-vectors sampled $i.i.d$ from $p(X,Y)$ with correlation defined such that $d_H(x^n, y^n) \leq d_{corr}$ where d_{corr} represents the distance between vectors with a certain correlation [79]. Throughout we assume the corner point of the Slepian-Wolf region with code rates $(R_x, R_y) = (H(X|Y), H(Y))$. In this case, we can assume that the sequence y^n is completely known at the decoder.

The Hamming code $(n,k) = (2^m - 1, 2^m - m - 1)$, $m \geq 3$, can achieve the corner point of the region when $H(X|Y) = H(Y|X) = m$ as in this case $R_x = m$. For this code, the set of codewords are partitioned into $2^{n-k} = 2^m$ cosets each containing 2^k valid codewords. The important property of Hamming code is that it partitions the space of codewords into cosets in which all the codewords are at least $d_{min} \geq d_{corr}$ far apart. This property in addition to the fact that the sequences x^n and y^n are correlated such that $d_H(x^n, y^n) \leq d_{corr}$, guarantees that an error-free decoding of the sequences is possible at the decoder.

The sequence x^n is compressed using a (n,k) linear block parity check code $\mathcal{C}$ with rate $R_x = \frac{n-k}{n}$ with the parity check matrix H. In particular, x^n is compressed into the 2^{n-k} syndromes (associated with cosets each with size 2^k). The code is *complete* in the sense that the union of its cosets contains all the n-vector sequences. The conditions for the code to achieve the corner point of the SW region is:

$$R_x = \frac{n-k}{n} \geq H(X|Y) \;\rightarrow\; k \leq n(1 - H(X|Y))$$

The syndrome is computed by $s^{(n-k)} = x^n H^T$ and transmitted to the decoder. H is a full rank parity check matrix with the form $H = [B|A]$ where A and B are $(n-k)\times(n-k)$ and $(n-k)\times(k)$ binary matrices, respectively. The matrix $H' = A^{-1}H = [A^{-1}B|I]$, a systematic parity check matrix, is used to compute the syndrome.

ML Detection: When the search through the codewords of a bin is too complex

to be performed exhaustively, maximum likelihood syndrome decoding can be used. For this purpose, suppose syndrome s^j is received. We need to associate an error vector with the received syndrome. A simple way to do that is to consider $w^j = [0_{1\times k}|s^j]$ (hence $w^j H^T = s^j$, c.f. property stated in lemma 3.3.2). Therefore we need to decode:

$$x^n = (c^i \oplus w^j),$$

where $c^i \in C$ is the codeword in *original coset* corresponding to x^n, i.e. the coset containing the channel codes (corresponding to syndrome $s = 0$). From this and the fact that $y^n = x^n + w^n$, we have:

$$y^n = c^j \oplus w^j \oplus w^n \rightarrow y' = y^n \oplus w^j = c^i \oplus w^n,$$

where we omit superscript n for a vector y' for clarity. Now, using y', the codeword c^j can be detected using ML detection (we refer to the detected codeword $\hat{c}^i$). The detected sequence is in the original coset, hence its counterpart in the coset with leader s^j is obtained by $\hat{x}^i = \hat{c}^i \oplus \hat{w}^j$.

Remark 3 *As can be seen from this example, compared to the Hamming channel code, the main difference here is that the information vectors b^k are no longer of any importance in the sense that now the codewords consist of all n-bit vectors in n-dim space(e.g. for Hamming $(7,4)$ code there are 2^7 valid codewords).*

Remark 4 *The main assumption in DISCUS (and generally codes based on Wyner's correlation channel idea) is that the correlation information between the two sequences is known. If fact, the success of DISCUS depends upon the proper choice of H (or G) with respect to the hypothetical correlation channel between X and Y. In other words, the code defined by H should be a* good code *for this channel [104]. For instance, the $(7,4)$ Hamming code is a good code for distributed coding of pairs generated by a binary symmetric source with correlation parameter such that the sequences are always closer than $d_H = 1$.*

3.5 Distributed estimation using EM

We extend the DISCUS algorithm, described above, to distributed estimation of the source parameter ρ using SI. The proposed scheme is in fact an implementation of the EM algorithm for MLE of the source parameter using side–information. The encoding and decoding operations are defined as follows:

3.5.1 Encoder

The compressed sequence u^m is computed as the syndrome of a linear block parity check code with the parity check matrix $H \in \{0,1\}^{m \times n}$ with rate $R_x = m/n$ defined as:

$$u^m = x^n H^T. \tag{3.12}$$

The sequence x^n is simply compressed using a $(n, n-m)$ linear block parity check code with the parity check matrix H. The encoder computes the syndrome of the coset containing the input sequence x^n. Provided that $R_x \geq R_c$, where R_c is the available capacity for transmission channel, the sequence u^m can be assumed to be perfectly available at the decode/estimator.

3.5.2 Decoder/Estimator

Given the SI y^n and the syndrome u^m, it is desired to estimate ρ. We define the coset corresponding to the compressed sequence u^m as:

$$\mathcal{C}(u^m) \triangleq Coset(u^m, H) = \{x^n : u^m = x^n H^T\}.$$

The MLE of ρ using the EM algorithm consists of the following two consecutive steps:

3.5.3 Expectation step (E-step)

In expectation step t, it is assumed that the parameter ρ is known (or initialized in the first iteration) and fixed, i.e. ρ_t . The expectation of the likelihood with respect to the posterior distribution of x^n given y^n is computed. The expectation is over those input vectors that could be responsible for generation of the syndrome u^m. For this purpose, in iteration t, the function $\mathcal{F}$ introduced in (3.9) is computed as follows:

$$\begin{aligned}\mathcal{F}(\rho|\rho_t) &= \sum_{x^n \in \mathcal{C}(u^m)} p(x^n|y^n;\rho_t) \log p(y^n, x^n;\rho) \\ &= E_{\rho_t} \log p(y^n, x^n;\rho) \\ &= E_{\rho_t}\Big[\log p(y^n|x^n;\rho) p(x^n;\rho)\Big] \end{aligned} \tag{3.13}$$

where for any function $l(x^n)$, E_{ρ_t} is the expectation with respect to the posterior distribution defined as:

$$\begin{aligned} E_{\rho_t} l(x^n) &= \sum_{x^n \in \mathcal{C}(u^m)} p(x^n|y^n;\rho_t) l(x^n) \\ &= \sum_{x^n \in \mathcal{C}(u^m)} \frac{p(y^n|x^n;\rho_t)p(x^n;\rho_t)}{\sum_{x^n \in \mathcal{C}(u^m)} p(y^n|x^n;\rho_t)p(y^n;\rho_t)} l(x^n). \end{aligned} \tag{3.14}$$

Here, we used the definition of the likelihood from (3.3) for $p(y^n|x^n;\rho_t)$, and used the abbreviated notation $d = d_H(x^n, y^n)$. All summations are over all the sequences in coset $\mathcal{C}(u^m)$.

By assuming a uniform sampling of input sequences x^n in each coset, the prior information is *independent* of the parameter ρ, e.g. $p(x^n;\rho) = \frac{1}{2^k}$. By substitution of the likelihood defined in (3.3) the $\mathcal{F}$ function becomes:

$$\begin{aligned}\mathcal{F}(\rho|\rho_t) &= E_{\rho_t}\Big[\log p(y^n|x^n,\rho) + \log\frac{1}{2^k}\Big] \\ &= E_{\rho_t}\Big[\log(\rho^{d(x^n,y^n)}(1-\rho)^{(n-d(x^n,y^n))}) + \log\frac{1}{2^k}\Big] \\ &= E_{\rho_t}\Big[d(x^n,y^n)\log(\rho) + (n-d(x^n,y^n))\log(1-\rho) + \log\frac{1}{2^k} - k\Big] \end{aligned} \tag{3.15}$$

3.5.4 Maximization step (M–step)

In the M–step, the expectation of the likelihood function represented by the $\mathcal{F}$ function that is computed in E–step is maximized and a new value for the parameter ρ_{t+1} is obtained:

$$\begin{aligned} \hat{\rho}_{(t+1)} &= \arg\max_{\rho} \mathcal{F}(\rho|\rho_t) \\ &= \frac{1}{n} E_{\rho_t} d(x^n, y^n), \end{aligned} \tag{3.16}$$

where the expectation is defined in (3.14).

By choosing a proper initial value for ρ, (3.16) provides an iterative solution for the MLE of ρ. According to this equation, the MLE of ρ in (3.5) is averaged over all input vectors in the coset designated by the syndrome u^m. The weights used in computation of the expectation are posterior probabilities of each individual sequence given the SI y^n. The iterations continue until convergence.

Remark 5 *In contrast to distributed coding, sufficiently long sequences are required for efficient distributed estimation. The MLE is not accurate (and generally not consistent nor efficient) for short sequences. On the other hand, for a (n, k) linear code, the size of each coset is (2^k) and hence the computational complexity of (3.14) grows exponentially with k (also with n). For instance, for a code (50, 25, 25) code with $R_x = 0.5n = 25$ bits this number is $2^{25} = 33,554,432$.*

More importantly, as k grows, it becomes more computationally challenging to locate all the sequences that belong to a desired coset. In fact, the assignment of sequences to each bin becomes an untractable operation for large n.

Therefore, for (asymptotically) consistent and efficient estimation, (i.e. with a large n) the use of low-complexity codes becomes inevitable. We continue to develop a low–complexity extension of the ideas presented above using LDPC codes.

Note: Before presenting a low–complexity implementation of the EM algorithm over factor–graphs, we review some background including LDPC codes, message passing over a factor–graph and application of LDPC codes for distributed coding. These subjects are covered to the extent as needed for our purposes. More detail can be found in the classical references. Then we extend the algorithms presented here to LDPC codes. The algorithm is a low-complexity implementation of source–coding with SI using LDPC *coset codes* equipped with the EM algorithm for correlation parameter estimation.

3.6 Low density parity check codes

LDPC codes were first discovered by Robert Gallager 40 years ago in his Ph.D. dissertation [42]. These codes are best described by their parity-check matrix H [51]. The parity-check matrix H of a binary LDPC code has a small number of ones (hence is low density). The random structure of H provides the randomness that was predicted to be necessary for achieving Shannon's coding limits.

LDPC decoding is best explained by its factor graph (*aka* Tanner graph [62])– a bipartite graph with two sets of nodes (Figure 3.2). Each row of the parity check matrix $H_{(m\times n)}$ corresponds to a parity check equation, and a "1" in the row constrains the corresponding input bit (variable) to the check equation. Similarly, each column of H corresponds to an input variable (bit), and a "1" in the column corresponds to a check equation that the input bit is engaged in. For H there are m parity check equations constraining n variables in n-vector codewords. Therefore, there are n *variable nodes*, v, representing the variable bits. Also there are m *check nodes*, c, representing the parity check equations. Hence, for each "1" in $H(i,j)$ there exists an edge from variable node i to check node j. The number of edges is controlled by the density of "1"s. An LDPC with density γ $(0 \leq \gamma \leq 1)$ has *row-weight* $w_r = \gamma.n$, i.e.

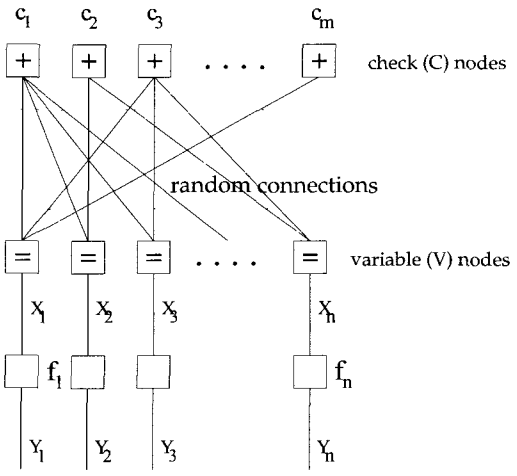


Figure 3.2: A bipartite graph for the LDPC codes

the number of "1"s in each row. Thus, there are w_r variables engaged in each parity check equation. Also, the *column-weight* is $w_c = \gamma.m$, i.e. the number of "1"s in each column. Thus, there are w_c parity check equations on each of the variables. When the number of "1"s are equal in all rows and also the number of "1"s are equal in all columns, the LDPC code is called *regular*, otherwise it is called *irregular*. In general, for a given rate and for the same code lengths, the performance of optimized irregular codes is expected to be superior to that of regular codes [80]. In this chapter, we only consider regular LDPC codes.

The main reason for using a bipartite graph for representing LDPC codes is that it allows the application of the *message passing* algorithm [62] for decoding. In what follows we describe the decoding algorithm for LDPC codes by means of message passing over a bipartite factor graph.

Remark 6 *The reminder of this chapter is presented for maintaining the continuity. The materials are borrowed (with minor changes) from [84] and [89]. Readers familiar with the subject can skip this section.*

3.6.1 LDPC Codes– Decoding Algorithm:

In addition to inherent randomness of the LDPC codes (implemented by the random structure of the parity check matrix H), the existence of an efficient decoding algorithm is the main reason for the success of these codes. The LDPC decoding algorithm, also referred to as *belief propagation* (BP) decoding algorithm, is a member of message passing iterative decoding algorithms [62]. In each iteration the probability (likelihood) of each variable bit is propagated over the graph of the code, from variable nodes to check nodes and from check nodes back to variable nodes. The main idea is to approximate the exponentially complex maximum likelihood detection algorithm (see Section 3.3.2) into a sub–optimal factorized solution. The factorization of the likelihood is possible by observing that for an $i.i.d$ sequence of outputs from a noisy channel, the most likely sequence can be found over the lattice of codes by a local search. The local search is performed by passing the belief (likelihood) of variables over the graph while continuously validating the decoded sequence by check nodes. More precisely, the messages passed from a variable node x to a check node c is the probability that x has a certain value ("0" or "1") given the observed value of that variable node, and all the values communicated to x in the prior round from the check nodes connected to x other than c itself. On the other hand, the message passed from c to x is the probability that x has a certain value ("0" or "1") given all the messages passed to c in the previous round from message nodes other than x itself.

The messages over the LDPC bipartite graph can be derived for two main cases, *a posteriori probability* (APP) and the *log-likelihood ratio* (LLR) defined below.

Suppose sequence y^n is received from a noisy channel with input x^n. For binary variables X and Y we define the likelihood functions are assumed as the following:

- $L(x) = \frac{\Pr(x=0)}{\Pr(x=1)}$ the likelihood ratio of x.
- $L(x|y) = \frac{\Pr(x=0|y)}{\Pr(x=1|y)}$ the conditional likelihood ratio of x given y,

- $LL(x) = \log L(x)$, the log-likelihood ratio of x,
- $LL(x|y) = \log L(x|y)$ the conditional log-likelihood ratio of x given y,

It is also assumed that the LDPC code is a "channel code". Thus, the parity check equations restrict the decoded sequence x^n to be in the original coset, i.e. all check nodes need to be zero.

Remark 7 *When the LDPC decoding is used for* syndrome decoding *(as it will be in distributed estimation or coding), this assumption need to be modified. The modification is such that non-zero syndrome (or check equations) be possible. More detail will be presented in the following sections.*

We begin with the following definitions:

- The variable nodes (v-nodes) are associated with random variables X_i , $1 \leq i \leq n$.
- The check nodes (c-nodes) are associated with the check equations c_j, $1 \leq j \leq m$.
- $V_j = \{$v-nodes connected to c-node c_j $\}$
- $V_{j\backslash i} = \{$v-nodes connected to c-node c_j $\} \setminus \{X_i\}$ [2]
- $C_i = \{$c-nodes connected to v-node X_i $\}$
- $C_{i\backslash j} = \{$c-nodes connected to v-node X_i $\} \setminus \{$ c_j $\}$
- $M_v(\sim i) = \{$messages from all v-nodes except node X_i $\}$
- $M_c(\sim j) = \{$messages from all c-nodes except node $c_j\}$

[2]The operator $\setminus$ means "excluding"

- $P_i = \Pr(x_i = 1|y_i)$
- S_i =the event that the check equations involving X_i are satisfied,
- The messages from v-nodes to c-nodes $q_{ij}(b) = \Pr(x_i = b|S_i, y_i, M_c(\sim j))$, where $b \in \{0, 1\}$.
- The messages from c-nodes to v-nodes $r_{ji}(b) = \Pr(\text{check equation } c_j \text{ is satisfied}|x_i = b, M_v(\sim i))$, where $b \in \{0, 1\}$.

Remark 8 *In terminology used in message-passing over factor graphs, two random variables $\mu_{X_i \to c_j}$ and $\mu_{c_j \to X_i}$ represent messages from random variables X_i to check node c_j and vice versa, respectively. Having defined the messages q_{ij} and r_{ji} above, these messages for the APP, LR, and LLR algorithms respectfully are as follows:*

$$\begin{aligned} \mu_{X_i \to c_j} &= q_{ij}(b) \\ \mu_{c_j \to X_i} &= r_{ji}(b) \end{aligned}$$

$$\begin{aligned} \mu_{X_i \to c_j} &= \frac{q_{ij}(0)}{q_{ij}(1)} \\ \mu_{c_j \to X_i} &= \frac{r_{ji}(0)}{r_{ji}(1)} \end{aligned}$$

$$\begin{aligned} \mu_{X_i \to c_j} &= \log \frac{q_{ij}(0)}{q_{ij}(1)} \\ \mu_{c_j \to X_i} &= \log \frac{r_{ji}(0)}{r_{ji}(1)} \end{aligned}$$

Property 1: For an equiprobable random variable x using Bayes' rule we have $L(x|y) = L(y|x)$.

Property 2: For *i.i.d* random variables $y_1, ..., y_n$:

$$LL(x|y_1, ..., y_n) = \sum_{i=1}^{n} LL(x|y_i).$$

We need the following theorem due to Gallager [42] for APP message computation:

Theorem 3.6.1 *Consider a sequence of M independent binary digits a_i for which* $\Pr(a_i = 1) = p_i$. *Then the probability that $\{a_i\}_{i=1}^{M}$ contains an even number of "1"s is:*

$$\frac{1}{2} + \frac{1}{2}\prod_{i=1}^{M}(1 - 2p_i).$$

The following theorem proves the same result for the LLR case.

Theorem 3.6.2 *For binary RVs $x_1, ..., x_n$ and RVs $y_1, ..., y_n$ and a binary variable s_i:*

$$LL(x_1 + x_2 + ... + x_n = s_i|y_1, ..., y_n) = (1 - 2s_i)\log\frac{1 + \prod_{i=1}^{n}\tanh(\frac{l_i}{2})}{1 - \prod_{i=1}^{n}\tanh(\frac{l_i}{2})},$$

where $l_i = LL(x_i|y_i)$.

Proof 3.6.1 *See Appendix C*

This theorem shows a mechanism by which the parity check equations are enforced and hence reinforce the belief for the most likely codeword. For instance, when all s_i are set to zero, this theorem gives a probabilistic measure for the extent that a codeword belongs to the original coset. Other values of of s_i gives this information for other cosets.

3.6.2 Messages over the factor graph (APP case)

We can define the messages for the APP case as follows:

3.6.2.1 Messages from the variable nodes to the check nodes

$$\begin{aligned} q_{ij}(0) &= \Pr(x_i = 0 | y_i, S_i, M_c(\sim j)) \\ &= (1 - P_i) \Pr(S_i | x_i = 0, y_i, M_c(\sim j)) / \Pr(S_i) \\ &= K_{ij}(1 - P_i) \prod_{j' \in C_{i \setminus j}} r_{j'i}(0) \end{aligned} \tag{3.17}$$

Similarly:

$$q_{ij}(1) = K_{ij} P_i \prod_{j' \in C_{i \setminus j}} r_{j'i}(1), \tag{3.18}$$

where constants K_{ij} is chosen properly to ensure that $q_{ij}(0) + q_{ij}(1) = 1$.

Therefore the message from v-node X_i to c-node c_j is:

$$\mu_{X_i \to c_i}(x_i) = q_{ij}(x_i). \tag{3.19}$$

3.6.2.2 Messages from the check nodes to the variable nodes

Referring to the Theorem 3.6.2, we notice that p_i corresponds to $q_{ij}(1)$, since when $x_i = 0$, the bits $\{x'_i : i' \in X_{j \setminus i}\}$ must contain an even number of 1's in order to check node c_j to be satisfied, we have:

$$r_{ji}(0) = \frac{1}{2} + \frac{1}{2} \prod_{i' \in X_{j \setminus i}} (1 - 2q_{i'j}(1)) \tag{3.20}$$

$$r_{ji}(1) = 1 - r_{ji}(0). \tag{3.21}$$

Therefore the message from c-node c_j to v-node X_i is:

$$\mu_{c_i \to X_i}(b) = r_{ji}(b). \tag{3.22}$$

3.6.2.3 Initialization

Suppose $y_i \in \{0,1\}$ is observed at the factor node f_i as shown in Figure 3.2 as the output of the hypothetical $BSC(\rho)$ with probability of error $\rho = \Pr(y_i = b^c | x_i = b)$ for any bit $b \in \{0,1\}$. We have:

$$\begin{aligned} \Pr(x_i = b|y_i) &= \frac{p(y_i|x_i)p(x_i)}{p(y_i|x_i=0)p(x_i=0) + p(y_i|x_i=1)p(x_i=1)} \\ &= \begin{cases} 1-\rho & \text{when } y_i = b \\ \rho & \text{when } y_i = b^c. \end{cases} \end{aligned} \tag{3.23}$$

where we made use of the fact that for BSS, $p(x_i = 0) = p(x_i = 1) = \frac{1}{2}$.

The message passing algorithm (MPA) starts with initialization of probabilities from variable nodes X_i generated by the likelihood of observing y_i at node f_i. Therefore, the initialization for variable nodes is as follows:

$$q_{ij}(b) = \Pr(x_i = b|y_i) = \begin{cases} P_i & \text{for } b = 1 \\ 1 - P_i & \text{for } b = 0, \end{cases} \tag{3.24}$$

$\forall i,j$ for which $h_{ij} = 1$ and zero otherwise. Here we used the definition $P_i = \Pr(x_i = 1|y_i)$.

3.6.3 Summary of MPA (APP Case:)

- *Initialize:* According to Equation 3.24, given the received symbol y_i for all $0 \leq i \leq n-1$ compute $P_i = \Pr(x_i = 1|y_i)$. Set $q_{ij}(0) = 1 - P_i$ and $q_{ij}(1) = P_i$ for all i,j for which $h_{i,j} = 1$.
- *Iterate:* Update $\{r_{ji}(b)\}$ using Equations 3.20 and 3.21
- Update $\{q_{ij}(b)\}$ using Equations 3.17 and 3.18 and solve for the constants K_{ij}.
- For $0 \leq i \leq n-1$, compute:

$$Q_i(0) \triangleq K_i(1 - P_i) \prod_{j \in C_i} r_{ji}(0), \tag{3.25}$$

and

$$Q_i(1) \triangleq K_i P_i \prod_{j \in C_i} r_{ji}(1), \tag{3.26}$$

where constants K_i are chosen such that $Q_i(0) + Q_i(1) = 1$. Notice that Q_i is similarly defined as q_{ij} except that it is summed over all incoming messages.

- For $0 \leq i \leq n-1$, set

$$\hat{x}_i = \begin{cases} 1, & \text{if } Q_i(1) > Q_i(0); \\ 0, & \text{else.} \end{cases} \tag{3.27}$$

If $\hat{x^n} H^T = 0$ or the number of iterations exceeds a predefined number, stop, otherwise continue at the *iterate* step.

3.6.4 Messages over the factor graph (LLR case)

Due to the multiplicative nature of the messages, the APP message passing algorithm is numerically unstable. The log-domain version of the algorithm is following. Define:

$$L(x_i) = \log\left(\frac{\Pr(x_i = 0|y_i)}{\Pr(x_i = 1|y_i)}\right) \tag{3.28}$$

$$L(r_{ji}) = \log\left(\frac{r_{ji}(0)}{r_{ji}(1)}\right) \tag{3.29}$$

$$L(q_{ij}) = \log\left(\frac{q_{ij}(0)}{q_{ij}(1)}\right) \tag{3.30}$$

$$L(Q_i) = \log\left(\frac{Q_i(0)}{Q_i(1)}\right) \tag{3.31}$$

$$\tag{3.32}$$

3.6.4.1 Messages from variable nodes to check nodes

The message from v-node X_i to c-node c_j is:

$$\mu_{X_i \to c_j}(x_i) = \log \frac{q_{ij}(0)}{q_{ij}(1)}. \tag{3.33}$$

We divide Equation 3.17 and 3.18 we have:

$$L(q_{ij}) = L(x_i) + \sum_{j' \in C_i \backslash j} L(r_{j'i}) \tag{3.34}$$

Similarly,

$$L(Q_i) = L(x_i) + \sum_{j \in C_i} L(r_{ji}). \tag{3.35}$$

3.6.4.2 Messages from check nodes to variable nodes

The message from c-node c_j to v-node X_i is:

$$\mu_{c_j \to X_i}(x_i) = \log \frac{r_{ji}(0)}{r_{ji}(1)}. \tag{3.36}$$

To compute the LLR message, we first replace $r_{ji}(0)$ with $1 - r_{ji}(1)$ in (3.20) we have:

$$1 - 2r_{ji}(1) = \prod_{i' \in V_j \backslash i} (1 - 2q_{i'j}(1)).$$

We use $\tanh\left[\frac{1}{2}\log(p_0/p_1)\right] = p_0 - p_1 = 1 - 2p_1$ and re-write the above equation into:

$$\tanh\left(\frac{1}{2}L(r_{ji})\right) = \prod_{i' \in V_j \backslash i} \tanh\left(\frac{1}{2}L(q_{i'j})\right). \tag{3.37}$$

To simplify this expression, let:

$$\begin{aligned} L(q_{ij}) &= \alpha_{ij}\beta_{ij} \\ \alpha_{ij} &= \text{sign}[L(q_{ij})] \\ \beta_{ij} &= |L(q_{ii})| \end{aligned}$$

So (3.37) becomes:

$$\tanh\left(\frac{1}{2}L(r_{ji})\right) = \prod_{i' \in Vj \backslash i} \alpha_{i'j} \cdot \prod_{i' \in V_j \backslash i} \tanh\left(\frac{1}{2}\beta_{i'j}\right).$$

Then we have:

$$\begin{aligned} L(r_{ji}) &= \prod_{i'} \alpha_{i'j}.2\tanh^{-1}\left(\prod_{i'}\tanh\left(\frac{1}{2}\beta_{i'j}\right)\right) \\ &= \prod_{i'} \alpha_{i'j}.2\tanh^{-1}\log^{-1}\log\left(\prod_{i'}\tanh\left(\frac{1}{2}\beta_{i'j}\right)\right) \\ &= \prod_{i'} \alpha_{i'j}.2\tanh^{-1}\log^{-1}\sum_{i'}\log\left(\tanh\left(\frac{1}{2}\beta_{i'j}\right)\right) \\ &= \prod_{i'\in V_j\setminus i} \alpha_{i'j}.\phi\left(\sum_{i'\in V_j\setminus i}\phi(\beta_{i'j})\right), \end{aligned} \tag{3.38}$$

were we have defined:

$$\phi(x) = -\log\left[\tanh(x/2)\right] = \log\left(\frac{e^x+1}{e^x-1}\right),$$

which also $\phi^{-1}(x) = \phi(x)$ for $x > 0$.

3.6.4.3 Initialization

The initialization step for different channels is different. For the BSC(ρ) and a binary bit y_i observed, we have:

$$L(q_{ij}) = L(x_i) = (-1)^{y_i}\log\left(\frac{1-\rho}{\rho}\right). \tag{3.39}$$

3.6.5 Summary of MPA (LLR Case:)

- *Initialize:* According to ((3.39), given the received symbol y_i for all $0 \le i \le n-1$ compute $L(q_{ij})$ for all i, j for which $H_{i,j} = 1$.
- *Iterate:* Update $\{L(r_{ji})\}$ using (3.38
- Update $\{L(q_{ij})\}$ using ((3.34)
- Update $\{L(Q_i)\}$ using (3.35)

- For $0 \leq i \leq n-1$, set

$$\hat{x}_i = \begin{cases} 1, & \text{if } L(Q_i) < 0; \\ 0, & \text{else.} \end{cases} \tag{3.40}$$

If $\hat{x^n} H^T = 0$ or the number of iterations exceeds a predefined number, stop, otherwise continue at the *iterate* step.

3.7 Low-complexity distributed estimation

We now extend the algorithm presented in Section 3.5 to LDPC codes and message passing over factor graphs. Factor graphs have been used to implement the EM algorithm previously in [38] for the purpose of joint channel estimation and symbol detection, and has been used for detection of symbols from a binary symmetric source in [39]. In [33] the EM algorithm is described in terms of message passing on factor graphs. The implementation presented here is essentially similar to the latter reference. Nevertheless, appropriate changes are applied to tailor the method for our purposes.

The factor graph for the ML estimation with side–information is depicted in Figure 3.3(b). The LDPC part of the graph is a standard LDPC code for *coset codes*. The input sequences x^n in each coset $Coset(u^m, H)$ are protected by the LDPC channel code. In other words, the check equations characterized by the parity check matrix H force the inputs to be in the coset $Coset(u^m, H)$ specified by the syndrome u^m.

The main idea implemented by the algorithm consists of the following steps:

- *Initialization:* First the parameter ρ is initialized and fixed. This parameter is used to compute the likelihood functions existing in the LDPC decoding scheme. A proper initialization, in general, plays an important role in the success of the EM-based algorithms. Despite this fact, as is shown in the simulations, when

code rates chosen properly, the success of the proposed algorithm does not depend on the initialization.

- *LDPC syndrome decoding:* Given a received sequence y^n and syndrome sequence u^m, multiple cycles of LDPC syndrome decoding are performed. As will discussed in detail, the parity check conditions in LDPC decoding are modified such that non–zero syndrome sequences (corresponding to cosets other than the original coset) are supported by the LDPC decoding scheme.

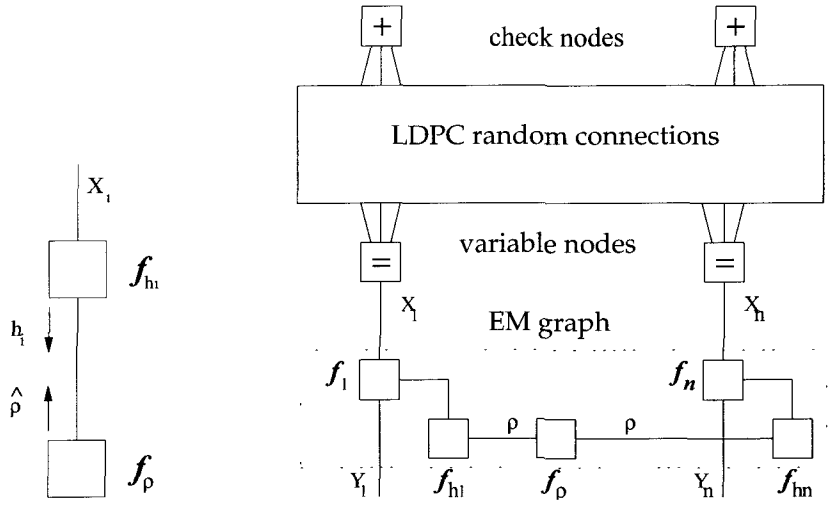
- *EM steps:* The result of the soft decoding of x^n in LDPC syndrome decoding is used to compute the posterior distribution present in the E–step. The soft decoding probabilities are in fact the posterior probability of each bit x_i given the received sequences y^n and u^m. These probabilities are treated as prior knowledge about each bit to compute the posterior probabilities in this step.

- The maximization of the M–step follows which computes a new estimate for the parameter ρ.

- *Convergence:* Unless a convergence is reached, the new estimated parameter is fed back to the LDPC factor graph for another round of LDPC syndrome decoding.

Therefore, the algorithm alternates between three consecutive cycles, i.e. the computation of posterior probabilities through the LDPC syndrome decoding; the computation of the expectation of the likelihood function using these posterior probabilities; and the maximization of the computed expectation to obtain a new estimate of the parameter ρ. The final estimate $\hat{\rho}$ is computed when the algorithm converges.

The graph related to the EM algorithm is depicted in Figure 3.3(a). In the following, each of the above-mentioned steps as well as corresponding individual messages over the factor–graph are explained. We first study the necessary modifications to



(a) a (b) b

Figure 3.3: (a) EM factor graph and corresponding messages, (b) EM-LDPC Factor Graph for ML estimation with side–information

LDPC decoding in order to implement LDPC syndrome decoding, i.e. when non–zero syndrome sequences are permitted. Then a detailed derivation of the E and M steps of the EM algorithm is presented.

Remark 9 *In the following discussion all summations are over* $x^n \in \mathcal{C}(u^m) = Coset(u^m, H)$.

3.7.1 LDPC–based syndrome decoding for distributed coding

The application of LDPC codes for distributed coding is a straightforward generalization of DISCUS (see [66] and [104] and the references therein). In order that the LDPC codes can be used for distributed estimation, an important change should be

made. In channel codes, the only valid coset is the one with *all–zero* syndrome, also referred to as the *original coset*. This coset contains the channel codewords for the particular choice of the generator matrix (and its corresponding parity check matrix).

Remark 10 *The compressed sequence u^m in distributed coding is computed as the syndrome of an LDPC code. Therefore, when clear from context, we refer to syndrome by s^m and u^m, interchangeably.*

In the original derivations by Gallager [42], all check variables were set to $s_i = 0$, meaning that the parity check equations enforced *even parity* in the codewords. In contrast, as we noticed in DISCUS, in distributed coding all cosets even with *non–zero* syndromes are valid. Therefore, when belief propagation (BP) for syndrome decoding in distributed source coding is desired, the parity check equations should be changed properly to take into account the non–zero syndrome bits. More specifically, for those non–zero syndrome bits, the roles of "(b = 0)" and "(b = 1)" are interchanged in messages. More details follow.

3.7.1.1 Messages from check nodes to variable nodes- APP case

Referring to Theorem 3.6.2, we notice that p_i corresponds to $q_{ij}(1)$, since when $x_i = 0$, the bits $\{x'_i : i' \in V_{j\backslash i}\}$ must contain an "*even*" number of 1's in order for the check node c_j to be satisfied. For the case of a non–zero syndrome bit, i.e. $s_i = 1$, the corresponding check node should be one. This is equivalent to saying that the bits $\{x'_i : i' \in V_{j\backslash i}\}$ must contain an "*odd*" number of 1's in order for the check node c_j to be satisfied. This change may be implemented by interchanging the role of the messages $r_{ij}(0)$ and $r_{ij}(1)$ for those check nodes whose syndrome bits are $s_i = 1$ in Equations 3.20 and 3.21, thus:

$$\begin{aligned} r_{ji}(1) &= \frac{1}{2} + \frac{1}{2} \prod_{i' \in V_{j\backslash i}} (1 - 2q_{i'j}(0)) \\ r_{ji}(0) &= 1 - r_{ji}(1). \end{aligned}$$

3.7.1.2 Messages from check nodes to variable nodes- LLR case

For the check nodes with a non–zero syndrome, i.e. $s_i = 1$, the changes made in the APP case, correspond to a sign change in the LLR. Hence (3.38) becomes:

$$L(r_{ji}) = (1 - 2s_i) \prod_{i' \in V_j \backslash i} \alpha_{i'j} \phi \left(\sum_{i' \in V_j \backslash i} \phi(\beta_{i'j}) \right),$$

This is previously used in literature [85] [66].

3.7.2 The E-Step

We begin with the following two facts:

Likelihood Factorization: Since the samples are *i.i.d*, the likelihood function can be factorized:

$$p(y^n|x^n;\rho) = \prod_{i=1}^{n} p_i(y_i|x_i;\rho). \tag{3.41}$$

Posterior Factorization: It is easy to verify that for *i.i.d.* samples, the expectation of the likelihood $p(y_i|x_i;\rho)$ with respect to the posterior distribution, at iteration t, can be marginalized as follows:

$$\begin{aligned} E_{\rho_t} \log p(y_i|x_i;\rho) &= \sum_{x^n \in \mathcal{C}(u^m)} p(x^n|y^n;\rho_t) \log p_i(y_i|x_i;\rho) \\ &= \sum_{x^n \in \mathcal{C}(u^m)} p_i(x_i|y_i;\rho_t) \log p_i(y_i|x_i;\rho) \\ &= \sum_{x^n \in \mathcal{C}(u^m)} E_i \log p_i(y_i|x_i;\rho), \end{aligned} \tag{3.42}$$

where E_{ρ_t} is defined in Equation 3.14 and E_i is the expectation with respect to the posterior distribution $p_i(x_i|y_i;\rho_t)$.

Using these two properties, we continue with what was derived in Section 3.5.3

(see (3.13)):

$$\begin{aligned}\mathcal{F}(\rho|\rho_t) &= \sum_{x^n \in \mathcal{C}(u^m)} p(x^n|y^n;\rho_t) \log p(y^n, x^n;\rho) \\ &= E_{\rho_t} \log p(y^n, x^n;\rho) \\ &= E_{\rho_t} \log p(y^n|x^n;\rho) p(x^n;\rho) \\ &= \sum_{x^n \in \mathcal{C}(u^m)} E_i \Big[\log p_i(y_i|x_i;\rho) + \log p(x_i;\rho) \Big]. \end{aligned} \tag{3.43}$$

For computing the expectation operations E_i in the above equations, the posterior distributions $p_i(y_i|x_i;\rho)$ are needed. In the following, by we show that these probability values can be received as the results of the LDPC syndrome decoding.

The E–step is started by receiving a message from node f_i towards the node f_{hi}. It can be seen from Figure 3.3(b) that the branches connecting the these pair of nodes correspond to random variables X_i. Therefore, at the beginning of each E-step, the node f_i implements its function which is computing the posterior distribution of x_i given the message received from the LDPC syndrome decoding algorithm ($\mu_{X_i \to f_i}$ or equivalently Q_i in (3.25)) and the received symbols y_i as the following:

$$p_i(x_i|y_i;\rho_t) = \frac{p(y_i|x_i;\rho_t).\mu_{X_i \to f_i}}{\sum_{x_i} p(y_i|x_i;\rho_t).\mu_{X_i \to f_i}} \tag{3.44}$$

where in this stage of operations (iteration t), we assumed that the node f_i assumed a known value ρ_t for the parameter and implements Bayes' rule for computing the posterior probabilities. For thus purpose, the messages received from the LDPC syndrome decoding are treated as the prior probability of each symbol x_i. For these messages, we have used the standard *a posteriori* probability (APP) messages $\mu_{X_i \to f_i}(0)$ and $\mu_{X_i \to f_i}(1)$ generated from node X_i. Note that, for the APP case, these messages correspond to Q_i's- the computed marginalized posterior probability of X_i's after sufficient

number of iterations in the LDPC decoding defined in (3.25) repeated here:

$$\mu_{X_i \to f_i}(0) = Q_i(0) = K_i(1-P_0)\prod_{j\in C_i} r_{ji}(0) \tag{3.45}$$

$$\mu_{X_i \to f_i}(1) = Q_i(1) = K_i P_0 \prod_{j\in C_i} r_{ji}(1), \tag{3.46}$$

where $P_0 = \Pr(x_i = 1|y_i)$ and C_i is the set of edges connected from the check nodes to variable node i, and r_{ji} is the message received at variable node i from check node j. The constants K_i are chosen properly such that $\mu_{X_i \to f_i}(0) + \mu_{X_i \to f_i}(1) = 1$ (see (3.17) and (3.18)).

The likelihood functions in (3.44) can be simplified into:

$$p_i(y_i|x_i = 0; \rho_t) = (\rho_t)^{y_i}(1-\rho_t)^{y_i^c} \tag{3.47}$$

$$p_i(y_i|x_i = 1, \rho_t) = (\rho_t)^{y_i^c}(1-\rho_t)^{y_i}, \tag{3.48}$$

where y_i^c is the *complement* of the binary variable y_i.

Now that the expectation operators have been computed, the function node f_{hi} computes the likelihood function for each element i defined in the following (see 3.3(a)):

$$h_{f_{hi}\to\rho} \triangleq h_i(y_i, \rho) \triangleq E_i \log p_i(y_i|x_i; \rho). \tag{3.49}$$

By this definition, the E-step objective function can be written as:

$$\begin{aligned}\mathcal{F}(\rho|\rho_t) &= \sum_{x^n \in \mathcal{C}(u^m)} E_i\Big[\log p_i(y_i|x_i;\rho) + \log p(x_i;\rho)\Big] \\ &= \sum_{x^n \in \mathcal{C}(u^m)} h_i(y_i,\rho) + \log p(x_i;\rho),\end{aligned} \tag{3.50}$$

which is the sum of the messages receiving from the nodes f_{hi}. Note as it was stated previously, since the sampling over sequences in each coset is performed uniformly, the distribution $p(x_i;\rho)$ is independent of ρ and does not contribute to the maximization of $\mathcal{F}$ with respect to ρ. Thus this term can be ignored in the M–step.

3.7.3 The M-Step

In the M–step (see 3.10), the maximization is with respect to ρ:

$$\begin{aligned}
\rho_{(t+1)} &= \arg\max_{\rho} \mathcal{F}(\rho|\rho_t) \\
&= \arg\max_{\rho} \sum_{x^n \in \mathcal{C}(u^m)} E_i\Big[\log p_i(y_i|x_i;\rho) + \log p(x_i;\rho)\Big] \\
&= \arg\max_{\rho} \sum_{x^n \in \mathcal{C}(u^m)} E_i\Big[\log p_i(y_i|x_i;\rho)\Big] \\
&= \arg\max_{\rho} \sum_{x^n \in \mathcal{C}(u^m)} E_i\Big[z_i \log\rho + (1-z_i)\log(1-\rho)\Big], \qquad (3.51)
\end{aligned}$$

where we substituted (3.3) for $p_i(y_i|x_i;\rho)$ in the above. Here $z_i = x_i \oplus y_i$ is a binary sum of the random variable x_i and variable y_i.

Remark 11 *Using the definition from (3.49), the M–step involves solving the following optimization:*

$$\rho_{(t+1)} = \arg\max_{\rho} \sum_i h_i(y_i,\rho). \qquad (3.52)$$

Therefore, the optimization objective is the sum of individual messages $h_{X_i \to f_\rho}$ (see Figure 3.3(a).

The solution to the optimization (3.51) is:

$$\rho_{(t+1)} = \sum_i E_i(z_i). \qquad (3.53)$$

Remark 12 *The upward message along the edge ρ from function node f_ρ is the result of the M–step optimization and therefore a new estimate $\hat{\rho}^{t+1}$ defined as (see Figure 3.3(a)):*

$$\rho_{f_\rho \to \rho} = \rho_{(t+1)}.$$

This message arrives intact to the nodes f_i through the nodes f_{h_j} in order to set a new value for the likelihood function implemented by these nodes (see (3.23)).

Remark 13 *Once the posterior probabilities in (3.44) are computed, the expectation in (3.53) can be simplified into:*

$$\begin{aligned}\rho_{(t+1)} &= \sum_i E_i(z_i) \\ &= \sum_i p_i(x_i = 0|y_i)y_i^c + p_i(x_i = 1|y_i),\end{aligned}$$

where again y_i^c is the complement of y_i.

3.7.4 Scheduling the Decoding/Estimation

Due to possible existence of short cycles in the LDPC coset code, we choose proper scheduling through which multiple cycles of decoding are performed before a new value for the parameter ρ is computed. Then, the result of the soft decoding of x^n is used to compute the posterior distribution used in the E–step. The maximization of the M–step follows. A new value of the parameter ρ obtained in the M–step is fed back to the LDPC factor graph for another round of LDPC decoding. Therefore, the algorithm alternates between three consecutive cycles, i.e. the computation of posterior probabilities through the LDPC syndrome decoding; the computation of the expectation of the likelihood function using these posterior probabilities; and the maximization of the computed expectation to obtain a new estimate of the parameter ρ. The final estimate $\hat{\rho}$ is computed when the algorithm converges.

Remark 14 *The final value of the parameter may be used in a typical distributed coding scheme to decode the syndrome u^m into the input sequence x^n. We will study this in simulations.*

3.7.5 The algorithm

- *Initialize:* Set an initial value for the parameter ρ. Given the received symbol y_i for all $0 \leq i \leq n-1$ compute $L(q_{ij})$ for all i, j for which $H_{i,j} = 1$. The

initial value for the parameter ρ affects this initialization through (3.24) for the APP case or (3.39) for the LLR case. These equations compute the posterior probability of x_i given the received y_i which is in turn a function of the likelihood of y_i given x_i (through Bayes' theorem). This likelihood is a function of the parameter ρ (see (3.23))

- *LDPC syndrome decoding:* Update $\{L(r_{ji})\}$ using Equation 3.38.

- Update $\{L(q_{ij})\}$ using Equation 3.34.

- Update $\{L(Q_i)\}$ using Equation 3.35.

- For $0 \leq i \leq n-1$, set

$$\hat{x}_i = \begin{cases} 1, & \text{if } L(Q_i) < 0; \\ 0, & \text{else.} \end{cases}$$

If $\hat{x^n} H^T = u^m$, where u^m is the received coset syndrome, or the number of iterations exceeds a predefined number (e.g. 300), continue with the E–step, otherwise continue with the steps for updating $\{L(r_{ji})\}$.

- *E-Step:* Compute the posterior probabilities in (3.42) for all i, required for the expectation operation.

- *M-Step:* Update the parameter ρ by solving (3.54).

- *Convergence:* If the difference between the new parameter value and the value for the previous iteration is less than a pre–defined threshold, stop the iteration. Otherwise, set the new parameter in (3.39) and re–start the LDPC decoding.

3.8 Region of Achievable Rates

In distributed estimation, a pair of rates (R_x, R_y) is called achievable if there exists a pair of encoders and a decoder/estimator that attain the same accuracy in distributed estimation as when the estimation is performed locally. In other words, when a pair of rates is achievable, distributed estimation can do as well as local estimation, i.e. it achieves the *local* Cramer-Rao lower bound (CRLB) [48].

In general, determination of the region of achievable rates for distributed estimation is not an easy and obvious task. In contrast to the Slepian–Wolf region in distributed coding [90], the region for distributed estimation is in general a function of the unknown parameter as well as the selected test channels [48] and hence difficult to determine. Nevertheless, for the case of the $BSS(\rho)$, the following theorem provides practical guidelines for choosing proper rates rather than precise determination of the region.

Theorem 3.8.1 *Suppose SI, y^n, is available perfectly at the decoder. For the $BSS(\rho)$ with $0 < \rho < \rho'$ where $0 < \rho' < 0.5$, if $R_x \geq H(\rho')$, ρ can be transmitted without loss of information, that is attain the same variance when uncompressed sequences x^n and y^n can be observed. The Fisher information is given by:*

$$J(\rho, R_x) \geq \begin{cases} \frac{1}{\rho(1-\rho)} \frac{R_x}{H(\rho')}, & R_x < H_b(\rho') \\ \frac{1}{\rho(1-\rho)}, & \text{otherwise.} \end{cases} \tag{3.54}$$

where $H_b(\rho) = -(\rho \log_2 \rho + (1-\rho)\log_2(1-\rho))$.

Proof 3.8.1 *See [88].*

According to this theorem, a sufficient statistic for ρ can be transmitted perfectly with a rate at least equal to $H(\rho')$. Moreover, when the rate is not sufficient, the theorem determines an upper bound on attainable Fisher information, i.e. a lower bound on estimation accuracy.

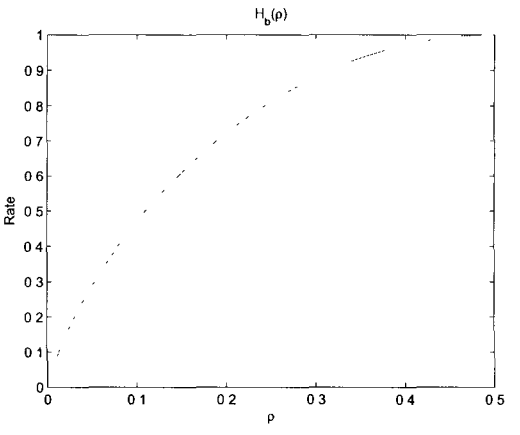


Figure 3.4: L = 6 iterations of the EM algorithm for estimation of $\rho = 0.1$ with random initial value (shown here for $\rho_0 = 0.4$, $n = 200$, $R_x \in [0.5, 1.0]$

Table 3.1: The binary entropy of binary symmetric source $BSS(\rho)$ for different values of parameter ρ.

ρ	0.05	0.10	0.15	0.20	0.25
$H_b(\rho)$	0.2864	0.4690	0.6098	0.7219	0.8113
ρ	0.30	0.35	0.40	0.45	0.50
$H_b(\rho)$	0.8813	0.9341	0.9710	0.9928	1.00

Figure 3.4 depicts $H_b(\rho)$. According to the above theorem, for any rate greater than $H_b(\rho)$, the local CRLB is achievable. As an example, for $\rho = 0.15$, a rate as large as $R_x \geq H_b(\rho) = 0.6098$ is required for distributed estimation to achieve the local CRLB index $(n/J(\rho, R_x) = \rho(1-\rho) = 0.1275n)$, where n is the number of samples (here the code length). In the following table, sufficient rates for achieving the local CRLB for different values of ρ are given. These rates are used in the simulations:

3.9 Simulations

The algorithm begins with an initial value for the unknown parameter ρ_0. Then it alternates between syndrome decoding, the E step and the M step. The E step is implemented by a sufficient number of iterations of the LDPC syndrome decoding (e.g. $300 - 500$). Due to the possible existence of short cycles in the LDPC coset code, we choose proper scheduling [62] through which multiple cycles of decoding are performed before the posterior probabilities are used in the E and M steps. Then the soft posterior probability value of the variables is used to implement the expectations in the E and M-steps. These decoding/estimation iterations are continued until convergence in the estimation of ρ is achieved or a maximum number of iterations is passed.

For studying the behavior of the estimation algorithm, a regular Gallager code is used, i.e. all the parity check matrices have 3 ones per column [42]. For generating the LDPC parity check matrix, we run through the graph trying to eliminate cycles of length 4; i.e., situations where pairs of rows share 1s in a particular pair of columns. The results are for $M = 1000$ Monte-Carlo runs of the algorithms.

The first set of results are for estimation of different values of parameter, i.e. $\rho = \{0.1, 0.15, 0.2, 0.25, 0.3, 0.4, 0.5\}$. Assuming an LDPC parity check matrix $H \in \{0,1\}^{m \times n}$, the graphs are for different values of available rates, $R_x = m/n = \{0.5, ..., 1.0\}$. The simulations are for a comparatively medium code length $n = 200$.

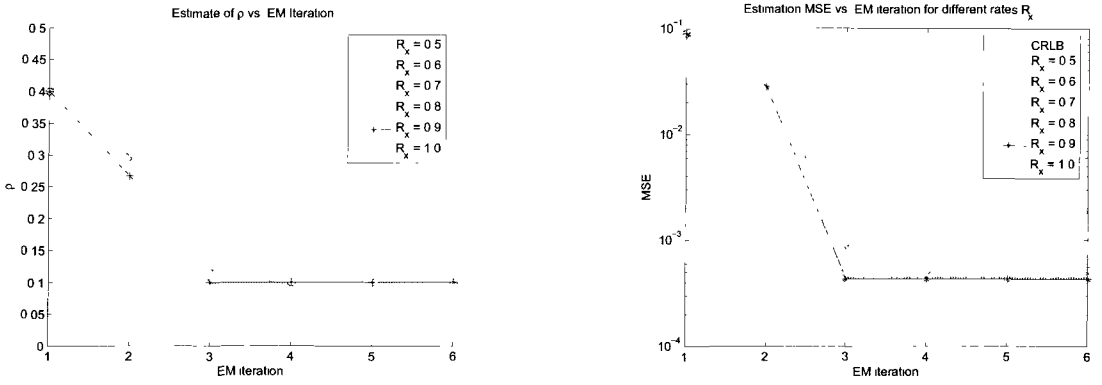
Figures 3.5(a) to 3.8(a) show the estimation results and Figures 3.5(b) to 3.8(b) show their corresponding mean square error (MSE) graphs for $L = 6$ iterations of the EM algorithm. Also, Figures 3.9(a) to 3.10(b) show the MSE of the converged EM algorithm versus different values of available rates, i.e. $R_x \in [0.5, 1.0]$. The CRLB for each case is computed as the inverse of the Fisher information by setting $\rho' = \rho$ in (3.54).

It can be seen from these graphs that for a rate $R_x \geq 0.7$, the algorithm successfully estimates the parameter in a few iterations and the MSE achieves the CRLB. However, for rates $R_x < 0.7$ estimation accuracy is not close to the bound. Longer code lengths and optimized irregular codes may be needed to achieve this bound.

Considering that the parameter is initialized randomly to a value between $0 \leq \rho_0 < 1/2$, the results suggest that provided a sufficient rate is chosen, the final estimation does not depend on initialization.

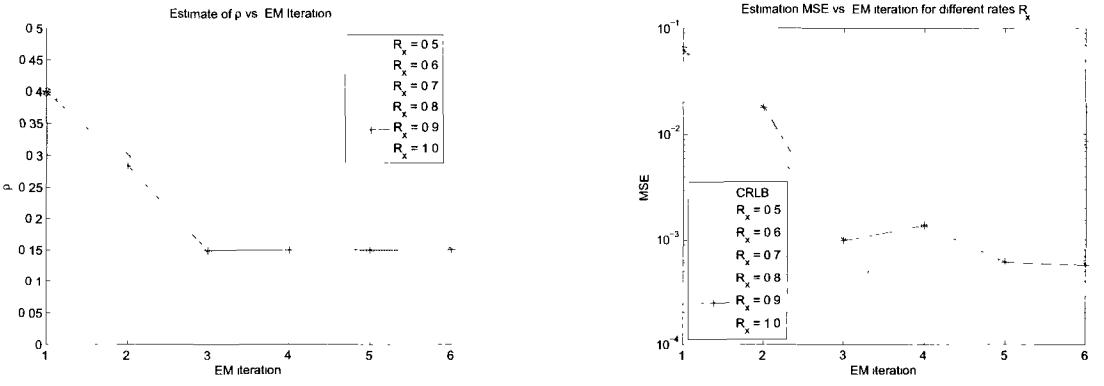
The algorithm is studied for different code lengths. It can be seen in Figures 3.11(a) and 3.11(b) that for $\rho = 0.10$ with a sufficiently large rate ($R_x = 0.65$), increasing the code length improves the estimation accuracy. When the rate is not sufficient, i.e. $R_x = 0.65$ for $\rho = 0.20$, increasing the code length does not show any improvement in accuracy (see Figures 3.12(a) and 3.12(b)). However, with a sufficiently large rate ($R_x = 0.90$), the accuracy can be improved with increasing the code length for $\rho = 0.20$ (see Figures 3.13(a) to 3.13(b)).

In a second set of simulations, the estimated parameters in the above are used for distributed coding using SI when the correlation parameter ρ is *not* known *a priori* (see Figures 3.14(a) to 3.15(b)). For this purpose, the bit error rate (BER) in decoding of sequence x^n is computed for different values of available rates $R_x \in [0.5, 1.0]$. Each of the graphs shows the BER for a different step of the EM algorithm as the parameter is estimated. Multiple graphs in each figure show the improvement in decoding achieved in different steps of the EM–algorithm. The results are for four cases $\rho = \{0.05, 0.1, 0.2, 0.3\}$. One can see that the proposed algorithm improves the BER in each step and hence may be used for joint parameter estimation and sequence decoding in distributed estimation/coding using SI scenarios.



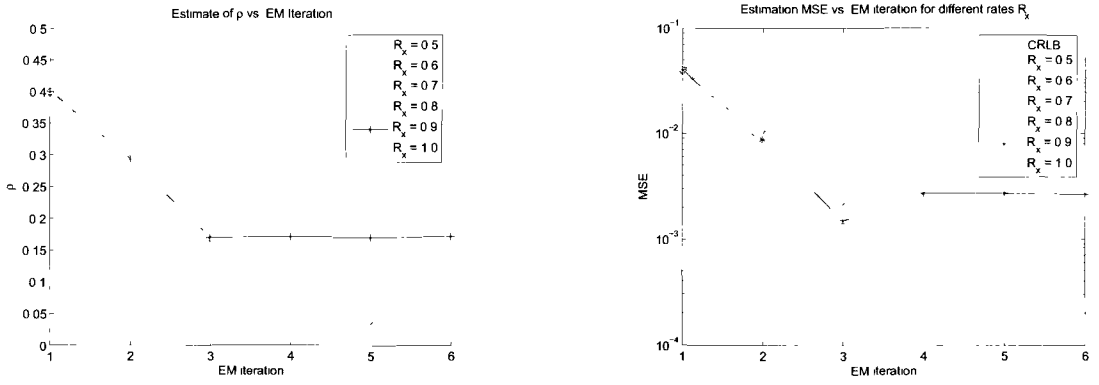
(a) (b)

Figure 3.5: (a) L = 6 iterations of the EM algorithm for estimation of $\rho = 0.10$ with random initial value, $n = 200$, $R_x \in [0.5, 1.0]$, and (b) MSE for L = 6 iterations of the EM algorithm, parameters similar to part (a)



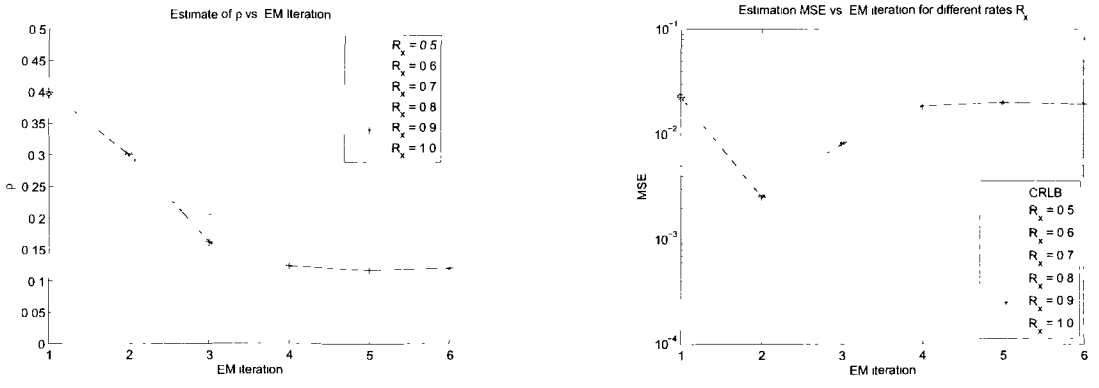
(a) (b)

Figure 3.6: (a) L = 6 iterations of the EM algorithm for estimation of $\rho = 0.15$ with random initial value, $n = 200$, $R_x \in [0.5, 1.0]$, and (b) MSE for L = 6 iterations of the EM algorithm, parameters similar to part (a)



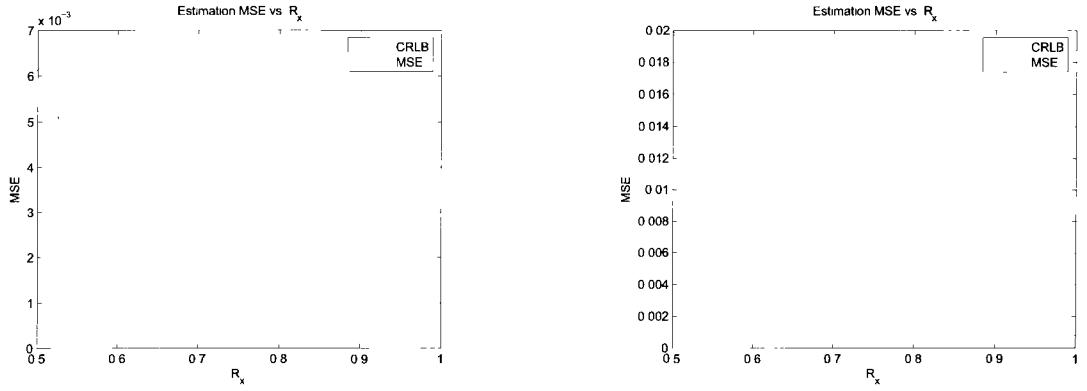
(a) (b)

Figure 3.7: (a) L = 6 iterations of the EM algorithm for estimation of $\rho = 0.20$ with random initial value (shown here for $\rho_0 = 0.4$, $n = 200$, $R_x \in [0.5, 1.0]$, and (b) MSE for L = 6 iterations of the EM algorithm, parameters similar to part (a)



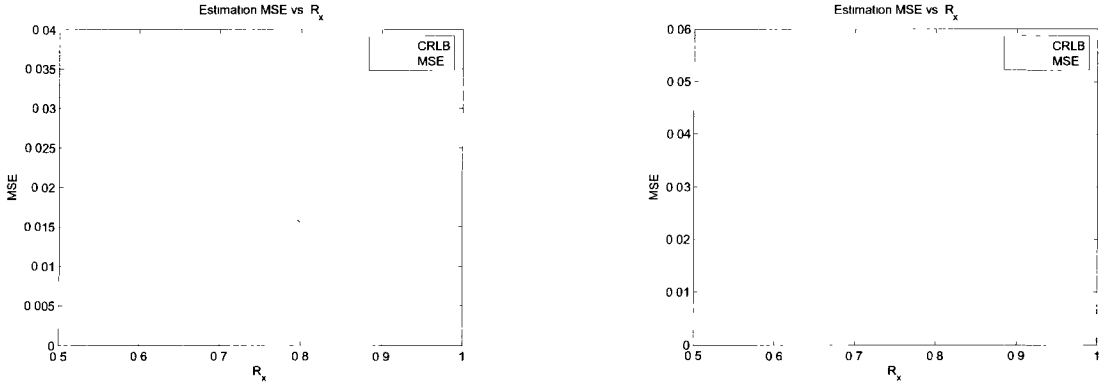
(a) (b)

Figure 3.8: (a) L = 6 iterations of the EM algorithm for estimation of $\rho = 0.25$ with random initial value, $n = 200$, $R_x \in [0.5, 1.0]$, and (b) MSE for L = 6 iterations of the EM algorithm, parameters similar to part (a)



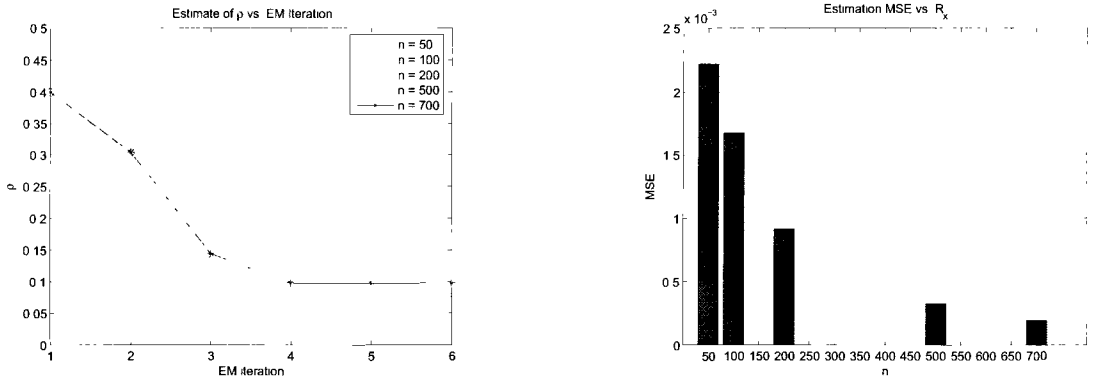
(a) (b)

Figure 3.9: (a) MSE for estimation of $\rho = 0.10$ after convergence of the EM algorithm for different available rates $R_x \in [0.5, 1.0]$ compared with the attainable CRLB (Eq. 3.54) (b) Similar to part (a) for $\rho = 0.15$



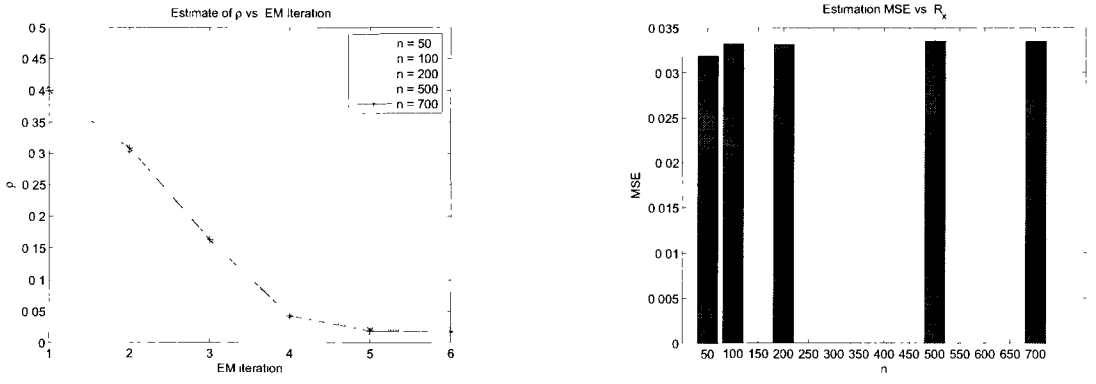
(a) (b)

Figure 3.10: (a) MSE for estimation of $\rho = 0.20$ after convergence of the EM algorithm for different available rates $R_x \in [0.5, 1.0]$ compared with attainable CRLB (Eq. 3.54) (b) Similar to part (a) for $\rho = 0.25$



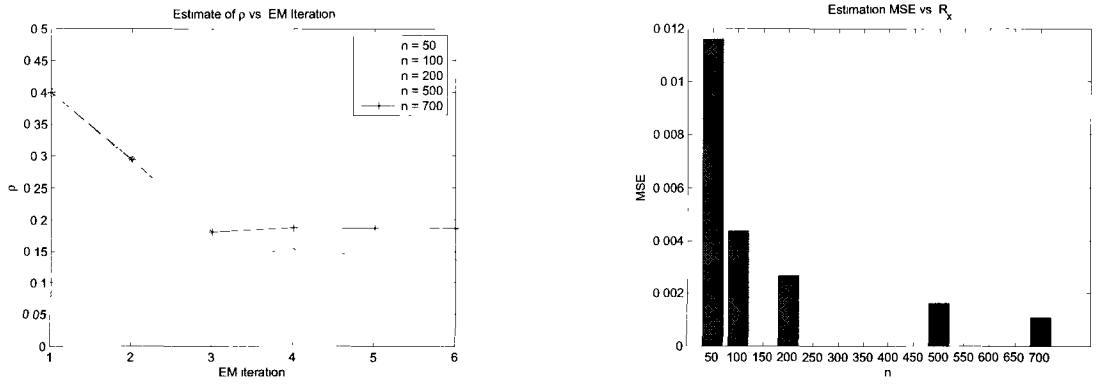
(a) (b)

Figure 3.11: (a) Estimation of $\rho = 0.10$ versus the EM iterations for different code lengths: Here $R_x = 0.65$ (b) MSE after convergence for different code lengths



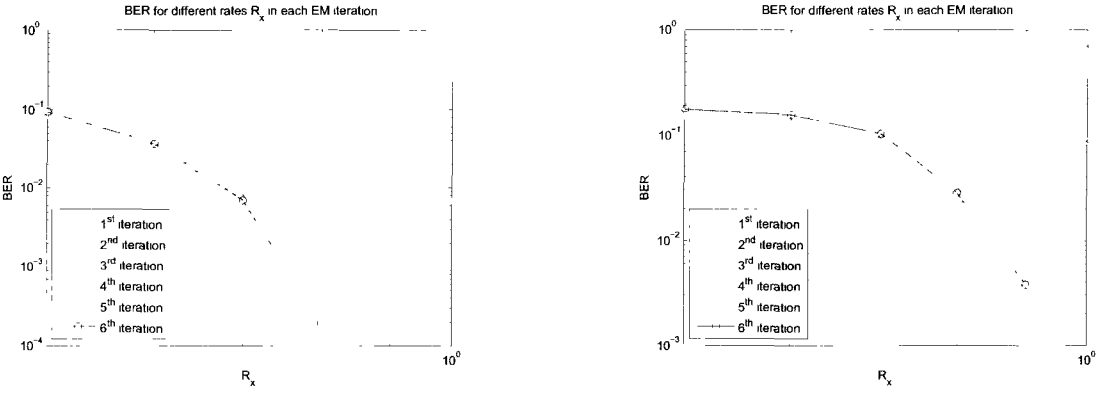
(a) (b)

Figure 3.12: (a) Estimation of $\rho = 0.20$ versus the EM iterations for different code lengths, here $R_x = 0.65$ (b) MSE after convergence for different code lengths



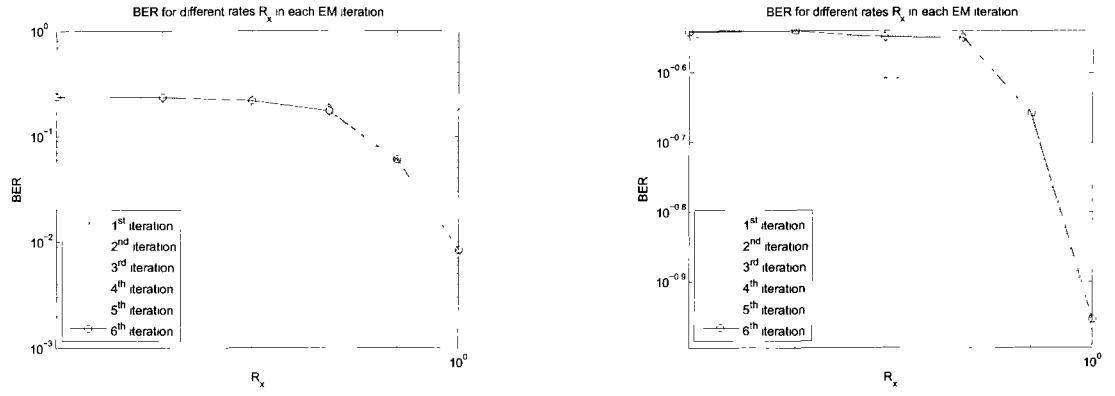
(a) (b)

Figure 3.13: (a) Estimation of $\rho = 0.20$ versus the EM iterations for different code lengths, here $R_x = 0.90$ (b) MSE after convergence for different code lengths



(a) (b)

Figure 3.14: (a) BER vs. available rates $R_x \in [0.5, 1.0]$ for different steps of the EM algorithm, here $\rho = 0.10$, and (b) Similar to part (b) for here $\rho = 0.15$



(a) (b)

Figure 3.15: (a) (a) BER vs. available rates $R_x \in [0.5, 1.0]$ for different steps of the EM algorithm, here $\rho = 0.20$, and (b) Similar to part (b) for here $\rho = 0.25$

3.10 Discussion

Extensive simulations, partly shown here, suggest that for comparably short code lengths the proposed algorithm is successful in the estimation of the BSS parameter provided that the rate is not too low ($R_x > 0.7$) and the BSS parameter is such that the correlation between the sequences is sufficiently high (i.e. $\rho \leq 0.25$). Interestingly, under these conditions, the algorithm was not dependent on the initial parameter chosen. The choice of the regular Gallager code with 3 ones per column was used to show the capability of the algorithm for estimation. This capability may be improved by using optimized codes, e.g. irregular codes with larger lengths.

The success of syndrome-based distributed coding schemes, e.g. DISCUS and LDPC-based codes, depends greatly on the knowledge of the underlying probability distribution at the decoder, based upon which a *good* correlation channel maybe chosen [104]. The existence of a universal block coding scheme accompanied with minimum entropy decoding is proved by Csiszar [30]. However, the construction of such codes is still an open research problem (for example refer to [22]). The proposed

algorithm may be considered as a first step towards designing such coding schemes, e.g. for achieving the corner points of the Slepian-Wolf region [90] when the underlying PD is not known perfectly.

Chapter 4

An Information Geometric Approach to MLE

4.1 Introduction

With regard to identification of multi-input multi-output (MIMO) communication channels, if a training set consisting of input-output pairs to the channel is available, then the unknown parameters can be estimated using an ML estimation method that incorporates training. However, there are situations in which the observations do not include the input signals, and therefore the estimation must be carried out using only the available output observations. In such cases, since the observations alone are incomplete for estimating the unknown model parameters, the identification process, usually referred to as *blind identification*, relies on the available structure of the input as well as the signal model assumed. Blind identification problem in this case is based only on the partially available data, otherwise known as the *incomplete data*. The EM algorithm [35] for solving the so-called *incomplete-data problem* is the main body of almost all algorithms that propose an approximate solution for stochastic blind identification. Previous work on the application of the EM algorithm

in communications is presented in [5, 40, 68, 105, 54, 26, 71, 100, 60, 41].

In this chapter we pose the incomplete data problem in an information geometric framework [28]. Information geometry encompasses a theoretical framework for a better understanding of estimation problems. The first paper that explicitly used the notion of information geometry for maximum likelihood estimation was due to Csiszar [27]. In this reference an iterative algorithm for minimizing the *Kullback-Liebler* (KL) distance between a given probability distribution representing the empirical distribution of the observations, and a family of probability distributions (the likelihood distributions) was proposed, and its relationship to ML estimation was investigated. Later in [32], an iterative algorithm for minimizing the KL-distance between two probability distribution (PD) convex sets was proposed and the application of the algorithm for maximum likelihood estimation was addressed. Maximum likelihood estimation with *incomplete data* was posed as a double minimization of the KL distance between two PD sets in [20]. A similar approach was used for learning in the Boltzman machine [19] and for iterative image reconstruction [18]. The same problem was considered as a double minimization of the KL-distance between two sets of PD's in [7]. Specifically in these references, it was shown that this double projection information geometric approach is closely related to the EM algorithm [7, 8]. Interested readers are encouraged to refer to [19] for a review of the two approaches.

The IGID algorithm uses a treatment similar to that given in [32]. The blind identification problem is implemented as a double minimization of the KL-distance between two PD sets. This minimization is realized in the form of iterative alternating projections. A major contribution here is that closed-form solutions for these projections are developed. This closed-form nature of the algorithm is made possible by a Gaussian assumption on the source distribution[1]. The primary advantage of the

[1]The impact of making such an assumption is discussed in Sect. IV.

proposed algorithm is computational. Previous EM algorithms for blind channel identification have not assumed Gaussianity of the source, and consequently suffer from computational complexity problems arising in the E-step, e.g., [5, 68, 105, 54, 26, 71]. In contrast, the closed-form analytical projections used by the proposed IGID algorithm reduce computational costs significantly, especially for large constellations, with minimal degradation in performance. Thus, the proposed method inherits the asymptotically optimal properties of ML stochastic estimation, at substantially reduced cost. A previous closed–form EM algorithm which uses the Gaussian assumption, for blindly identifying single–input, single–output channels is presented in [70].

Due to the similarity between information geometric alternating projection and EM algorithms, it would have been possible to derive a blind identification algorithm based on EM principles, that assumes a Gaussian source, instead of using information geometric principles. However, solving the blind identification problem in the information geometric framework gives new insight into the identification process and its relationship to the EM algorithm. Moreover, the development and the execution of the closed-form expressions for the required minimization (projection) operations are very straightforward and simple.

Notation: Bold upper-case (lower-case) symbols indicate a matrix (vector) quantity respectively, while a symbol in calligraphic style indicates a set of probability distributions. The notation $N(\boldsymbol{\mu}, \boldsymbol{\Psi})$ denotes a multivariate (complex) normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Psi}$. In this chapter, we consider joint distributions of the form $q(\boldsymbol{x}, \boldsymbol{y})$, and their associated marginal and conditional distributions $q(\boldsymbol{x})$ and $q(\boldsymbol{x}|\boldsymbol{y})$ respectively. Even though these are three distinct distributions, they are not denoted as such. The meaning of the distribution is evident from the structure of its argument. The subscript t is the IGID iteration index and k is the temporal index.

We refer to the following minimization

$$p^\star = \arg\min_{p\in\mathcal{P}} D(p||q)$$

as a type-I projection. It projects q onto $\mathcal{P}$, where q is an arbitrary PD, $\mathcal{P}$ is a set of PDs, and $D(p||q)$ is the $KL-$ distance measure, which is defined as

$$D(p||q) = \sum_x p(x)\log\frac{p(x)}{q(x)}$$

where p and q in this case are discrete PDs and $q(x) \neq 0$ over the range of x. We refer to the following minimization

$$q^\star = \arg\min_{q\in\mathcal{Q}} D(p||q)$$

as a type-II projection, which projects p onto a PD set $\mathcal{Q}$. Because the KL distance measure is asymmetric, these two projections have different characteristics [31]. In the sequel, we often do not indicate the type of projection we are referring to. The type is made clear from the context.

4.2 Stochastic ML Estimation

Here we consider the general problem of *stochastic* maximum likelihood (ML) estimation of parameters. The stochastic ML approach assumes a class of PD for the unknown variables and therefore has the advantage of imposing statistical structure on the inputs, in contrast to deterministic ML methods which ignore the statistics of the unknown quantities. For a comprehensive review of *deterministic* and *stochastic* methods to the blind identification problem, refer to [96].

We consider the following linear time-invariant MIMO system with M transmitters and N receivers:

$$\boldsymbol{y}(k) = \sqrt{\frac{\rho}{M}}\boldsymbol{H}\boldsymbol{x}(k) + \boldsymbol{v}(k) \tag{4.1}$$

where $\boldsymbol{y}(k) \in \mathbb{C}^N$ and $\boldsymbol{x}(k) \in \Omega^M$ are the output and the input vectors, respectively, k is the time index, and Ω is a complex constellation with C members, such that the average energy over all members of the constellation is unity. The quantity $\boldsymbol{H} \in \mathbb{C}^{N \times M}$ is the complex channel coefficient matrix, whose elements are zero mean random variables, scaled to unit *rms* values. The quantity ρ is the SNR on each receive channel. Also, the sources are chosen to be *i.i.d.*, whose components are zero–mean Gaussian. The quantity $\boldsymbol{v} \sim N(\boldsymbol{0}, \boldsymbol{\Psi})$ is the noise vector with generally unknown covariance $\boldsymbol{\Psi} \in \mathbb{C}^{N \times N}$. It is assumed that $\boldsymbol{\Psi}$ is full rank.

The above MIMO model is valid in an intersymbol–interference (*ISI*) free Rayleigh–fading channel. It is assumed the channel $\boldsymbol{H}$ and the covariance $\boldsymbol{\Psi}$ are constant over a block of L transmitted symbols. This model is useful in space–time coding systems, where in many cases it is desired to (semi) blindly identify the channel [4, 93, 94]. By doing so, a higher quality channel estimation is available for "mismatched" style detectors comparing to the case where a short training sequence is used. This provides a better performance with less training sequence and therefore higher rate [92]. This model is also widely adopted in OFDM systems, e.g., [5].

A joint *pdf* of the input and output variables, e.g. $q(\boldsymbol{z}; \boldsymbol{\theta})$, where $\boldsymbol{z} = [\boldsymbol{y}^T, \boldsymbol{x}^T]^T$ is the *complete* data, and $\boldsymbol{\theta} = (\boldsymbol{H}, \boldsymbol{\Psi})$ is the parameter set, provides a complete description of the underlying signal model. In general, for a given $\boldsymbol{z}$ there exists a one–to–one correspondence between $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ and $q(\cdot\,; \boldsymbol{\theta}) \in \mathcal{Q}$, where $\boldsymbol{\Theta}$ is the parameter space, and $\mathcal{Q}$ is the set of likelihood distributions, defined by

$$\mathcal{Q} = \{q(\boldsymbol{z}; \boldsymbol{\theta}), \boldsymbol{\theta} \in \boldsymbol{\Theta}\}. \tag{4.2}$$

The ML estimation task is then to choose a distribution in this family that best describes the complete data. By assuming that L independent complete data samples $\boldsymbol{z}_k$, $k = 1, \ldots, L$ are available, the maximum likelihood estimation problem is to find

the distribution $q^*(\boldsymbol{z};\boldsymbol{\theta}^*)$ that satisfies

$$q^*(\boldsymbol{z};\boldsymbol{\theta}^*) = \arg\max_{q\in\mathcal{Q}} \prod_{k=1}^{L} q(\boldsymbol{z}_k;\boldsymbol{\theta}). \tag{4.3}$$

There are situations where the complete data are only partially available, i.e., we observe only $\boldsymbol{y}$. In these circumstances, the question is how to maximize the likelihood of observations and select the distribution $q(\boldsymbol{z};\boldsymbol{\theta}) \in \mathcal{Q}$ given only the partially available data (also called *incomplete data*). Assuming that the input is discrete and distributed according to the *pdf* $p(\boldsymbol{x})$, one must solve the following equivalent *incomplete-data* problem:

$$q^*(\boldsymbol{y};\boldsymbol{\theta}^*) = \arg\max_{q\in\mathcal{Q}} \log \prod_{k=1}^{L} \sum_{\boldsymbol{x}_k} q(\boldsymbol{y}_k|\boldsymbol{x}_k;\boldsymbol{\theta}) p(\boldsymbol{x}_k). \tag{4.4}$$

4.2.1 The Information Geometric Approach to Stochastic ML Estimation

We recall here how the ML estimate of (4.4) can be re–written in the form of a projection onto a set of distributions. The discrete distribution case is presented here for simplicity (extension to the continuous case is straightforward). Assume that the domain of incomplete observations $\boldsymbol{y}$ is divided into J neighborhoods, Δy_j; $j = 1, ..., J$ with $\tilde{\boldsymbol{y}}_j$'s as their center points. Now, given the observations $\boldsymbol{y}_k$, $k = 1, \ldots, L$, we define the *empirical distribution* $\tilde{p}$ of the observations to be:

$$\tilde{p}(\tilde{\boldsymbol{y}}_j) = \frac{1}{L} \sum_{k=1}^{L} \delta(\tilde{\boldsymbol{y}}_j - \boldsymbol{y}_k) \; ; \quad j = 1, ..., J \tag{4.5}$$

where $\delta(\cdot)$ is the Kronecker delta function defined as:

$$\delta(\tilde{\boldsymbol{y}}_j - \boldsymbol{y}_k) = \begin{cases} 1 \; ; & \boldsymbol{y}_k \in \Delta \boldsymbol{y}_j \\ 0 \; ; & \text{otherwise} \end{cases} \tag{4.6}$$

Using $q(\boldsymbol{y}_k) = \sum_{\boldsymbol{x}_k} q(\boldsymbol{y}_k, \boldsymbol{x}_k)$, the MLE problem (4.4) can be written as

$$\begin{aligned}
q^* &= \arg\max_{q\in\mathcal{Q}} \log\Big(\prod_{k=1}^{L} q(\boldsymbol{y}_k)\Big) \\
&= \arg\max_{q\in\mathcal{Q}} \sum_{k=1}^{L} \log q(\boldsymbol{y}_k) && (4.7)\\
&= \arg\max_{q\in\mathcal{Q}} \sum_{j=1}^{J}\sum_{k=1}^{L} \delta(\tilde{\boldsymbol{y}}_j - \boldsymbol{y}_k) \log q(\boldsymbol{y}_k) && (4.8)\\
&= \arg\max_{q\in\mathcal{Q}} L\sum_{j=1}^{J} \tilde{p}(\tilde{\boldsymbol{y}}_j) \log q(\tilde{\boldsymbol{y}}_j) && (4.9)\\
&= \arg\max_{q\in\mathcal{Q}} \left\{ \sum_{j=1}^{J} \tilde{p}(\tilde{\boldsymbol{y}}_j) \log \tilde{p}(\tilde{\boldsymbol{y}}_j) - \sum_{j=1}^{J} \tilde{p}(\tilde{\boldsymbol{y}}_j) \log \frac{\tilde{p}(\tilde{\boldsymbol{y}}_j)}{q(\tilde{\boldsymbol{y}}_j)} \right\} \\
&= \arg\max_{q\in\mathcal{Q}} \{-\mathcal{H}(\tilde{p}) - D(\tilde{p} \parallel q)\} && (4.10)
\end{aligned}$$

where $\mathcal{H}(\tilde{p})$ is the entropy of the empirical distribution $\tilde{p}$. Eq. (4.8) is obtained from (4.7) by the definition of the Kronecker delta function. Eq. (4.9) follows from (4.8) by using the definition of the empirical distribution given in (4.5). Since the entropy of $\tilde{p}$, i.e. $\mathcal{H}(\tilde{p})$, does not depend on the variable of maximization in (4.7) the ML estimation problem becomes:

$$q^* = \arg\min_{q\in\mathcal{Q}} D(\tilde{p}(\boldsymbol{y}) \parallel q(\boldsymbol{y})). \tag{4.11}$$

Thus, the ML estimation problem is equivalent to finding the projection of $\tilde{p}(\boldsymbol{y})$ onto the set $\mathcal{Q}$.

Observe that the optimization of (4.11) must find the best joint distribution $q(\boldsymbol{y}, \boldsymbol{x}) \in \mathcal{Q}$ using only the information of the marginal distributions. To solve this incomplete-data problem we proceed according to the method of [20] and define $\mathcal{P}$ as the set of all possible empirical distributions whose marginal distribution over the unknown variable $\boldsymbol{x}$ is equal to the empirical distribution $\tilde{p}(\boldsymbol{y})$ of the observations:

$$\mathcal{P} = \{p(\boldsymbol{y}, \boldsymbol{x}) | \sum_{\boldsymbol{x}} p(\boldsymbol{y}, \boldsymbol{x}) = \tilde{p}(\boldsymbol{y})\}. \tag{4.12}$$

Now, for a given q_0, observe that:

$$\begin{aligned} D(p \parallel q_0) &= \sum_{\boldsymbol{y}} \sum_{\boldsymbol{x}} p(\boldsymbol{y}, \boldsymbol{x}) \log \frac{p(\boldsymbol{y}, \boldsymbol{x})}{q_0(\boldsymbol{y}, \boldsymbol{x})} \\ &= \sum_{\boldsymbol{y}} \sum_{\left\{\boldsymbol{x} \mid \sum_{\boldsymbol{x}} p(\boldsymbol{y}, \boldsymbol{x}) = \tilde{p}(\boldsymbol{y})\right\}} \tilde{p}(\boldsymbol{y}) p(\boldsymbol{x}|\boldsymbol{y}) \log \frac{\tilde{p}(\boldsymbol{y}) p(\boldsymbol{x}|\boldsymbol{y})}{q_0(\boldsymbol{y}) q_0(\boldsymbol{x}|\boldsymbol{y})}, \end{aligned} \tag{4.13}$$

where the last line follows because the joint distribution $p(\boldsymbol{y}, \boldsymbol{x})$ representing the observed data and the corresponding inputs is physically constrained to lie within $\mathcal{P}$; hence $p(\boldsymbol{y}, \boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y}) p(\boldsymbol{y}) = p(\boldsymbol{x}|\boldsymbol{y}) \tilde{p}(\boldsymbol{y})$. The above derivation can be extended as follows:

$$\begin{aligned} D(p \parallel q_0) &= \sum_{\boldsymbol{y}} \tilde{p}(\boldsymbol{y}) \log \frac{\tilde{p}(\boldsymbol{y})}{q_0(\boldsymbol{y})} + \sum_{\boldsymbol{y}} \tilde{p}(\boldsymbol{y}) \sum_{\{\boldsymbol{x} \mid \sum_{\boldsymbol{x}} p(\boldsymbol{y}, \boldsymbol{x}) = \tilde{p}(\boldsymbol{y})\}} p(\boldsymbol{x}|\boldsymbol{y}) \log \frac{p(\boldsymbol{x}|\boldsymbol{y})}{q_0(\boldsymbol{x}|\boldsymbol{y})} \\ &= D(\tilde{p}(\boldsymbol{y}) \parallel q_0(\boldsymbol{y})) + \mathrm{E}_{\tilde{p}(\boldsymbol{y})} D(p(\boldsymbol{x}|\boldsymbol{y}) \parallel q_0(\boldsymbol{x}|\boldsymbol{y})). \end{aligned} \tag{4.14}$$

Now, since q_0 and the empirical distribution of the observations $\tilde{p}(\boldsymbol{y})$ are given, the first term in (4.14) is unchanged by changing p. This term is thus a lower bound on the $KL-$distance between p and q_0. Therefore, the minimum $KL-$distance is achieved by letting $p(\boldsymbol{x}|\boldsymbol{y}) = q_0(\boldsymbol{x}|\boldsymbol{y})$ regardless of $\boldsymbol{y}$. This gives:

$$\min_{p \in \mathcal{P}} D(p \parallel q_0) = D(\tilde{p}(\boldsymbol{y}) \parallel q_0(\boldsymbol{y})) \quad \forall q_0 \in \mathcal{Q}. \tag{4.15}$$

Substitution of (4.15) in (4.11) gives:

$$\{q^*, p^*\} = \arg \min_{q \in \mathcal{Q}} \min_{p \in \mathcal{P}} D(p \parallel q). \tag{4.16}$$

Since the minimum $KL-$distance is achieved when $p(\boldsymbol{x}|\boldsymbol{y}) = q_0(\boldsymbol{x}|\boldsymbol{y})$, and by definition the marginal distribution of every distribution $p \in \mathcal{P}$ is equal to $\tilde{p}(\boldsymbol{y})$, one observes that $p^*(\boldsymbol{y}, \boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y}) \tilde{p}(\boldsymbol{y}) = q_0(\boldsymbol{x}|\boldsymbol{y}) \tilde{p}(\boldsymbol{y})$ achieves the minimum $KL-$distance in (4.15). This completes the proof for the following important theorem, which follows from [20]:

Theorem 1: Define the set $\mathcal{P}$ as in (4.12). Also define $\mathcal{Q}$ as the set of all likelihood PD's $q(\boldsymbol{y}, \boldsymbol{x}; \boldsymbol{\theta})$ each member of which is characterized by the parameter vector $\boldsymbol{\theta}$. The ML estimation of $\boldsymbol{\theta}$ can be obtained by the following double minimization:

$$\{q^\star, p^*\} = \arg \min_{q \in \mathcal{Q}} \min_{p \in \mathcal{P}} D(p \parallel q). \tag{4.17}$$

Also, the projection of a given likelihood distribution $q_0(\boldsymbol{y}, \boldsymbol{x})$ on $\mathcal{P}$ is given by:

$$p^*(\boldsymbol{y}, \boldsymbol{x}) = \arg \min_{p \in \mathcal{P}} D(p \parallel q_0(\boldsymbol{y}, \boldsymbol{x})) = q_0(\boldsymbol{x}|\boldsymbol{y})\tilde{p}(\boldsymbol{y}). \tag{4.18}$$

□

We therefore have the important result that stochastic ML estimation with incomplete data is equivalent to the double minimization of the $KL-$distance between the sets $\mathcal{Q}$ and $\mathcal{P}$. This double minimization is implemented using an iterative alternating projection method. After initializing with a suitable (p_0^*, q_0^*), at iteration t, $p_{t+1}^\star$ is the type-I projection of $q_t^\star$ onto $\mathcal{P}$, and $q_{t+1}^\star$ is the type-II projection of $p_{t+1}^\star$ onto $\mathcal{Q}$. Then, $t \leftarrow t + 1$ and the process iterates until convergence.

If these two sets of PD's are convex[2], the double minimization has a global minimum, convergence to which is guaranteed [31]. However, in our case, $\mathcal{Q}$ is not convex and therefore the proposed alternating projection algorithm requires that the initial estimate $q_0^\star$ be determined through a proper initialization (training) procedure to assist convergence to the global minimum. The convergence analysis of the alternating projections approach to ML estimation of incomplete data is analogous to that of the EM algorithm. With the latter, conditions to guarantee convergence to a global optimum are very difficult to establish [101]. Thus, appropriate training sequences are used to alleviate convergence of the EM algorithm to the global optimum. The same applies to the IGID algorithm.

As can be seen in the following section, the projection onto $\mathcal{Q}$ is a convex optimization that gives a unique solution which is guaranteed to be a member of the set

[2]For the definition of convexity, refer to [32].

of desired likelihood distributions.

It is straightforward to observe that:

$$\begin{aligned} D(p_{t+1} \parallel q_{t+1}) &\leq D(p_{t+1} \parallel q_t) \\ &\leq D(p_t \parallel q_t), \end{aligned} \tag{4.19}$$

Since $D(\cdot \parallel \cdot)$ is bounded from below by 0, the sequence of pdf's generated by the algorithm decreases monotonically in $KL-$distance, and convergence to a local minimum is guaranteed.

4.3 Application to Semi–Blind Channel Identification

In this section, we apply the above information geometric approach to develop the computationally efficient IGID algorithm for semi–blind ML estimation of a multiple–input, multiple–output (MIMO) channel. The method is *semi-blind* due to the fact that the initial point is obtained by training the algorithm in each data block. We show the exact equivalence of the IGID algorithm and the *variational* EM algorithm [82] in Appendix (D).

4.3.1 Signal Distributions

It is assumed that the input $\boldsymbol{x}(k) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Phi})$, where it is assumed $\boldsymbol{\mu} = \mathbf{0}$, and $\boldsymbol{\Phi}$ is assumed known. The set of likelihood distributions $\mathcal{Q}$ is parameterized by $\boldsymbol{\theta} = \{\boldsymbol{H}, \boldsymbol{\Psi}\}$, where $\boldsymbol{H}$ and $\boldsymbol{\Psi}$ are the channel, and the covariance matrix of the noise $\boldsymbol{v}$ in (4.1), respectively. Thus, each member of $\mathcal{Q}$ is a Gaussian likelihood distribution defined by:

$$q(\boldsymbol{z}; \boldsymbol{H}, \boldsymbol{\Psi}) = \mathcal{N}(\bar{\boldsymbol{z}}, \boldsymbol{Q}) \tag{4.20}$$

where:

$$\bar{\boldsymbol{z}} = \begin{bmatrix} \boldsymbol{H}\boldsymbol{\mu} \\ \boldsymbol{\mu} \end{bmatrix} \tag{4.21}$$

and

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{H}\boldsymbol{\Phi}\boldsymbol{H}^T + \boldsymbol{\Psi} & \boldsymbol{H}\boldsymbol{\Phi} \\ \boldsymbol{\Phi}\boldsymbol{H}^T & \boldsymbol{\Phi} \end{bmatrix} \tag{4.22}$$

where we have used the fact that $\boldsymbol{\Phi}^T = \boldsymbol{\Phi}$.

The expression for $\boldsymbol{Q}^{-1}$ is given for future convenience as (see Appendix (III):

$$\boldsymbol{Q}^{-1} = \begin{bmatrix} \boldsymbol{\Psi}^{-1} & -\boldsymbol{\Psi}^{-1}\boldsymbol{H} \\ -\boldsymbol{H}^T\boldsymbol{\Psi}^{-1} & \boldsymbol{\Phi}^{-1} + \boldsymbol{H}^T\boldsymbol{\Psi}^{-1}\boldsymbol{H} \end{bmatrix}. \tag{4.23}$$

For analytical and computational tractability, in the following we assume the empirical distribution corresponding to the observations is normally distributed; $\tilde{p}(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{r}, \boldsymbol{S})$, where $\boldsymbol{r}$ and $\boldsymbol{S}$ are the mean vector and the sample covariance matrix for the output observations, respectively. This is in contrast to the exact form of empirical distribution given by (4.5). We note that since the source is assumed zero mean, and if we assume the channel has finite zero frequency gain, then under these assumptions we have $\boldsymbol{r} = \boldsymbol{0}$. This fact is used later.

4.3.2 The First Projection: Computing the Best Complete-Data Distribution

Due to the Gaussian source assumption, the complete data distribution is jointly Gaussian. Thus, this task is equivalent to finding the best mean and covariance of this joint distribution. Having the distribution $q_t(\boldsymbol{y}, \boldsymbol{x} : \boldsymbol{\theta})$ obtained from the previous iteration, we now solve the first minimization:

$$p^* = \min_{p \in \mathcal{P}} D(p \parallel q), \tag{4.24}$$

which is a type-I projection of the given PD q on the PD set $\mathcal{P}$.

The unique solution is given by the second part of Theorem 1. Therefore, to obtain the optimum distribution, one needs to compute the joint distribution of the complete-data likelihood distribution. In the case of jointly Gaussian PD's, straightforward mathematical manipulations yield the following closed form solution:

$$p^* = q(\boldsymbol{x}|\boldsymbol{y})\tilde{p}(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{m}, \boldsymbol{P}^*), \tag{4.25}$$

where it can be shown that

$$\boldsymbol{m} = \bar{\boldsymbol{z}} + \boldsymbol{P}^* \overline{\boldsymbol{S}}^{-1} \begin{bmatrix} \boldsymbol{H}\boldsymbol{\mu} - \boldsymbol{r} \\ \boldsymbol{0} \end{bmatrix}, \tag{4.26}$$

$$(\boldsymbol{P}^*)^{-1} = \begin{bmatrix} \boldsymbol{\Psi}^{-1} - (\boldsymbol{H}\boldsymbol{\Phi}\boldsymbol{H}^T + \boldsymbol{\Psi})^{-1} + \boldsymbol{S}^{-1} & -\boldsymbol{\Psi}^{-1}\boldsymbol{H} \\ -\boldsymbol{H}^T\boldsymbol{\Psi}^{-1} & \boldsymbol{\Phi}^{-1} + \boldsymbol{H}^T\boldsymbol{\Psi}^{-1}\boldsymbol{H} \end{bmatrix} \tag{4.27}$$

and $\overline{\boldsymbol{S}}^{-1} \in \Re^{(M+N)\times(M+N)}$ is the covariance matrix $\boldsymbol{S}^{-1} \in \Re^{N\times N}$ properly augmented with zero blocks, i.e.:

$$\overline{\boldsymbol{S}}^{-1} = \begin{bmatrix} \boldsymbol{S}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}.$$

We note that, under the current assumptions, both $\boldsymbol{m}$ and $\bar{\boldsymbol{z}}$ are $\boldsymbol{0}$.

Therefore, to solve the first projection, i.e., to calculate $(\boldsymbol{P}^*)^{-1}$, it is sufficient only to modify the upper–left element of the inverse covariance matrix $\boldsymbol{Q}^{-1}$ given by (4.23), using the current estimates of $\boldsymbol{H}$ and $\boldsymbol{\Psi}$. The closed-form solution (4.27) avoids multi-dimensional integrations usually necessary in conventional EM-type algorithms. In Appendix I we show this projection is identical to the E-step of the EM algorithm.

4.3.3 The Second Projection: the Complete-Data ML Estimation

Given $p^*(\boldsymbol{z})$ from the previous projection, the second minimization is an ML estimation of the parameters, using the complete-data. The problem in the second

minimization is to find the best distribution in the likelihood PD set $\mathcal{Q}$ that fits the estimated complete-data. This is equivalent to finding the type-II projection of p^* onto the PD set $\mathcal{Q}$. Therefore, it is necessary to solve the following minimization problem:

$$q^* = \arg\min_{q\in\mathcal{Q}} D(p^* \parallel q). \tag{4.28}$$

Since the $\mathcal{Q}$ family is parameterized by $\boldsymbol{H}$ and $\boldsymbol{\Psi}$, the optimization is performed with respect to these parameters. Assuming the following block form for the covariance matrix $\boldsymbol{P}^*$ of the given distribution p^*

$$\boldsymbol{P}^* = \begin{bmatrix} \boldsymbol{P}_{11} & \boldsymbol{P}_{12} \\ \boldsymbol{P}_{12}{}^T & \boldsymbol{P}_{22} \end{bmatrix}, \tag{4.29}$$

the second projection is equivalent to the following minimization (see Appendix IV):

$$\begin{aligned} \{\boldsymbol{H}^*, \boldsymbol{\Psi}^*\} \;=\; & \arg\min_{\{\boldsymbol{H},\boldsymbol{\Psi}\}} \Big[\operatorname{trace}(\boldsymbol{\Psi}^{-1}\boldsymbol{P}_{11}) \\ & - 2\operatorname{trace}(\boldsymbol{\Psi}^{-1}\boldsymbol{H}\boldsymbol{P}_{12}^T) \\ & + \operatorname{trace}(\boldsymbol{\Phi}^{-1}\boldsymbol{P}_{22} + \boldsymbol{H}^T\boldsymbol{\Psi}^{-1}\boldsymbol{H}\boldsymbol{P}_{22}) \\ & - \log\det\boldsymbol{\Psi}^{-1} - \log\det\boldsymbol{\Phi}^{-1} \\ & - \log\det\boldsymbol{P}^* - d \Big], \end{aligned} \tag{4.30}$$

where $d = M + N$ is the dimension of the complete-data. Observe that by assuming a nonsingular noise covariance matrix $\boldsymbol{\Psi}$, the minimization is a convex optimization. It therefore has a unique solution. The minimization of the objective with respect to the parameters $\boldsymbol{H}$ and $\boldsymbol{\Psi}$ gives (see Appendix V):

$$\boldsymbol{H}^* = \boldsymbol{P}_{12}\boldsymbol{P}_{22}^{-1} \tag{4.31}$$

$$\boldsymbol{\Psi}^* = \boldsymbol{P}_{11} - \boldsymbol{P}_{12}\boldsymbol{P}_{22}^{-1}\boldsymbol{P}_{12}^T. \tag{4.32}$$

Therefore the iterative application of Equation (4.27), (4.31) and (4.32) generates the sequence of distributions $p_t, q_t, \;\; t = 0, 1, \ldots,$ which in the limit yield maximum

likelihood estimates for the model parameters $\boldsymbol{H}$ and $\boldsymbol{\Psi}$. It is shown in Appendix I that this projection is identical to the M-step of the EM algorithm.

4.3.4 Initialization Using Training

When a training data set consisting of input signal and output observation pairs are available, then identification is a *complete-data* ML estimation problem. Even though this problem is straightforward to solve in this case, it is interesting to note that it may be solved using an information geometric formulation. Maximum likelihood estimation corresponds to finding the closest (in the $KL-$distance sense) likelihood distribution $q(\boldsymbol{y}, \boldsymbol{x})$ to the empirical distribution of the input-output training data $\tilde{p}(\boldsymbol{y}, \boldsymbol{x})$:

$$q^* = \arg\min_{q \in \mathcal{Q}} D(\tilde{p} \parallel q) \tag{4.33}$$

Assume a set of L_{tr} training data pairs $\boldsymbol{z}_k = [\boldsymbol{y}_k, \boldsymbol{x}_k]^T, (k = 1, \ldots, L_{tr})$ is available. Assume that the empirical distribution (4.5) for the training data is modelled by a normal distribution, i.e. $\tilde{p}(\boldsymbol{z}) \sim \mathcal{N}(\boldsymbol{r}, \boldsymbol{S})$ where:

$$\boldsymbol{r} = \frac{1}{L_{tr}} \sum_{k=1}^{L_{tr}} \boldsymbol{z}_k$$

$$\boldsymbol{S} = \frac{1}{L_{tr}} \sum_{k=1}^{L_{tr}} \boldsymbol{z}_k \boldsymbol{z}_k^T = \frac{1}{L_{tr}} \sum_{k=1}^{L_{tr}} \begin{bmatrix} \boldsymbol{y}_k \boldsymbol{y}_k^T & \boldsymbol{y}_k \boldsymbol{x}_k^T \\ \boldsymbol{x}_k \boldsymbol{y}_k^T & \boldsymbol{x}_k \boldsymbol{x}_k^T \end{bmatrix}.$$

Then, it is straightforward to show that the estimates $\breve{\boldsymbol{H}}$ and $\breve{\boldsymbol{\Psi}}$ solving (4.33) are given by

$$\breve{\boldsymbol{H}} \sum_{k=1}^{L_{tr}} \boldsymbol{x}_k \boldsymbol{x}_k^T = \frac{1}{L_{tr}} \sum_{k=1}^{L_{tr}} \boldsymbol{y}_k \boldsymbol{x}_k^T \rightarrow \breve{\boldsymbol{H}} = \sum_{k=1}^{L_{tr}} \boldsymbol{y}_k \boldsymbol{x}_k^T \left(\boldsymbol{x}_k \boldsymbol{x}_k^T \right)^{-1}. \tag{4.34}$$

where $\boldsymbol{\Phi}$ is the source covariance matrix which is assumed known, and

$$\breve{\boldsymbol{\Psi}} = \frac{1}{L_{tr}} \sum_{k=1}^{L_{tr}} (\boldsymbol{y}_k - \breve{\boldsymbol{H}} \boldsymbol{x}_k)(\boldsymbol{y}_k - \breve{\boldsymbol{H}} \boldsymbol{x}_k)^T. \tag{4.35}$$

These results are similar to the least-squares solution for the ML estimation with training [99].

4.3.5 The IGID Algorithm: Summary

1. *Initialization*: Initial parameter estimates $\breve{\boldsymbol{H}}_0$ and $\breve{\boldsymbol{\Psi}}_0$ are obtained from the training data using (4.34) and (4.35) respectively.

2. *The first projection (onto $\mathcal{P}$), i.e., choosing the best empirical complete-data distribution:* In this step in iteration t, we compute the best distribution of the complete data $\boldsymbol{z} = [\boldsymbol{y}^T \boldsymbol{x}^T]^T$. Therefore we substitute the current values of $\boldsymbol{H}_t$ and $\boldsymbol{\Psi}_t$ into (4.27) to obtain the optimum covariance matrix $(\boldsymbol{P}^*)^{-1}$.

3. *The second projection (onto $\mathcal{Q}$), i.e., maximum likelihood estimation of parameters:* In this step, we compute the maximum likelihood estimate of the channel. Therefore, we invert $(\boldsymbol{P}^*)^{-1}$ and prepare the sub-blocks as in (4.29). Then we use Equations (4.31) and (4.32) to estimate the parameters $\boldsymbol{H}_{t+1}$ and $\boldsymbol{\Psi}_{t+1}$.

4. *Termination:* Check convergence by examining $D(p^* \parallel q^*)$ ((G.20)). If the distance is less than a predefined value ϵ, $0 < \epsilon \ll 1$, terminate; otherwise, continue with the first projection (Step 2).

4.3.6 Convergence of the IGID Algorithm

We have seen that if the PD sets $\mathcal{P}$ and $\mathcal{Q}$ are convex, then the IGID algorithm converges [31]. However, in our case, these PD sets are Gaussian and hence are not convex. Even though (4.19) guarantees non-increasing divergence, it does not guarantee convergence to a single point in the intersection of $\mathcal{P}$ and $\mathcal{Q}$. As with the conventional EM algorithm, analysis of convergence of the IGID algorithm to a global optimum is difficult and is beyond the scope of this thesis. Nevertheless, we

can demonstrate the behavior of the algorithm at convergence. In the following, we demonstrate there is a Gaussian distribution in $\mathcal{P}$ and one in $\mathcal{Q}$ that have equal means and covariances.

It is shown in Appendix VI that there exists a point $\{\hat{\boldsymbol{H}}, \hat{\boldsymbol{\Psi}}\}$ within $\mathcal{Q}$ such that

$$\boldsymbol{S} = \hat{\boldsymbol{H}}\boldsymbol{\Phi}\hat{\boldsymbol{H}}^T + \hat{\boldsymbol{\Psi}}, \tag{4.36}$$

where $\boldsymbol{S}$ is the covariance matrix of the observations $\boldsymbol{y}$ (defined in the paragraph under (4.23)). By substituting (4.36) into the upper–left block of (4.27) and substituting the values $\hat{\boldsymbol{H}}$ and $\hat{\boldsymbol{\Psi}}$, it is straightforward to show that $(\boldsymbol{P}^\star)^{-1}$ from (4.27) remains invariant from one iteration to the next. Further, $(\boldsymbol{P}^\star)^{-1}$ from (4.27) is equal to $\boldsymbol{Q}^{-1}$ from (4.23), when $\hat{\boldsymbol{H}}$ and $\hat{\boldsymbol{\Psi}}$ are substituted in (4.23). Thus, there exists a point in $\mathcal{P}$ and $\mathcal{Q}$ for which the covariance matrices of the distributions p_∞ and q_∞[3] are equal and invariant with iteration.

From Section 4.3.2, we have seen that the mean $\bar{\boldsymbol{z}}$ of q_∞ and the mean $\boldsymbol{m}$ of p_∞ are both zero and invariant with iteration.

Therefore, since the means and covariances of the distributions are equal and are invariant with iteration, then under the stated conditions, there exists a point within $\mathcal{P}$ and $\mathcal{Q}$ to which convergence is possible. Thus, the convergence region of the IGID algorithm includes at least one point.

A consequence of this property is that $D(p_\infty||q_\infty) \rightarrow 0$.

4.4 Simulations

4.4.1 Channel Estimation

In this section, we present simulation results for verifying the performance of the IGID algorithm for blind channel identification. The results are compared with a general

[3]Recall the subscript on p or q refers to iteration index.

previous EM-based algorithm which does not exploit a Gaussian source assumption, as summarized in Appendix B. ML estimation using all the data in the block as a training sequence is also performed, since this provides a lower bound on the performance of the algorithms. The IGID algorithm does not require the noise covariance to be known in general. However, in simulations we assume that the noise covariance matrix $\mathbf{\Psi}$ is known in order to be able to compare the results with previously reported algorithm. In each simulation, it is assumed that the channel gain matrix is constant within a block of length $L = 1000$ symbols.

In each block, the IGID and EM algorithms are initialized using training consisting of 10% of the block length, using (4.34) and (4.35). It is assumed that the symbols transmitted from each transmit antenna are selected uniformly from a 4-QAM (QPSK), 16-QAM or 64-QAM constellation, which has been normalized to unit variance. The values M and N are each chosen to be equal to 2. The receiver noise covariance matrix $\mathbf{\Psi}$ is set to the identity $\boldsymbol{I}$. The channel coefficient matrix $\boldsymbol{H}$ is scaled corresponding to the desired values of SNR, as in (4.1). The channel coefficient matrix itself is chosen so that its elements are *i.i.d.* circular complex Gaussian random variables, with zero mean and unit variance.

The first index we use for quantifying the performance of the algorithms is the root mean–squared (*rms*) error of the channel estimate, at each iteration of the algorithm. This index shows the speed of convergence of the algorithms, as well as the extent to which the algorithm is able to converge to the true channel gain matrices. For these experiments, two values of SNR, arbitrarily chosen to be 16dB and 6dB, are chosen to demonstrate the performance of the algorithms in typical high and low-SNR conditions.

Figure 4.1 shows the *rms* error of the channel gain matrix estimation for 50 Monte-Carlo runs, for the IGID algorithm (circle), the EM algorithm (star), and ML estimation using the entire block as a training sequence (dashed line), versus iteration

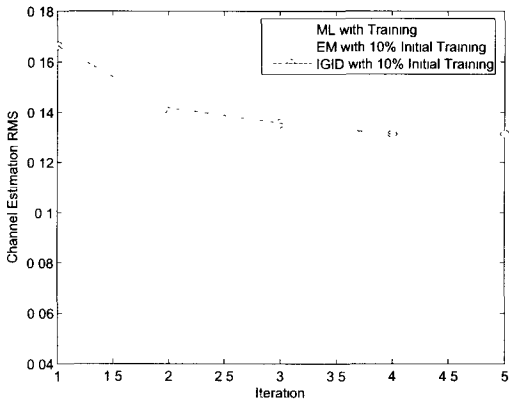


Figure 4.1: Convergence (*rms* error vs. iteration index) for channel gain estimation in the low-SNR regime (SNR=6dB) in a 2 by 2 MIMO communication system with 16-QAM modulation. "ML with Training" uses the whole block of data as a training sequence, whereas the EM and the IGID algorithms each use 10% of the data block for training. The error is evaluated over 50 Monte Carlo runs.

index, in the low-SNR regime ($SNR = 6$). The results are evaluated over $50,000$ symbols (50 blocks). It can be seen that the performance of the IGID algorithm is almost equal to that of the conventional EM algorithm. Also the IGID algorithm converges at a rate comparable to the EM-algorithm. Figure 4.2 shows the same results in the high-SNR regime ($SNR = 16$). Here, it can be seen that the IGID algorithm performance is only slightly degraded in terms of *rms* error compared to that of the EM algorithm.

4.4.2 Symbol-Error-Rate (SER)

To further examine the performance of the proposed algorithm, a symbol error rate (SER) analysis is performed. Each symbol is detected using an ideal ML procedure using the estimated values of $\boldsymbol{H}$ and $\boldsymbol{\Psi}$. Since in the simulations the number of sources is small, the usual complexity of ML detection is tractable in this case. The estimated channel gain matrix and the noise covariance matrix are used for ML detection of the

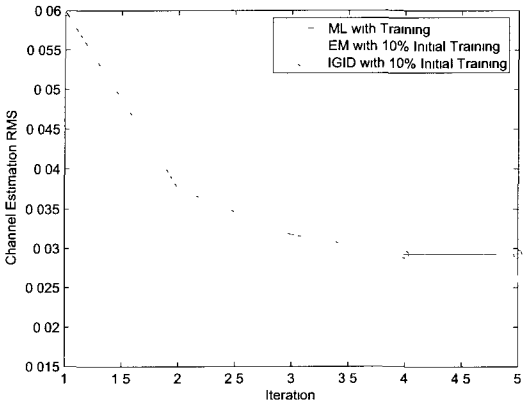


Figure 4.2: Same as Figure 4.1, except SNR = 16 dB

transmitted symbols $\boldsymbol{x}$, which are computed by:

$$\boldsymbol{x}^* = \max_{\boldsymbol{x}\in\Omega} p_x(\boldsymbol{x}|\boldsymbol{y}; \hat{\boldsymbol{H}}, \hat{\boldsymbol{\Psi}}) \tag{4.37}$$

where Ω is constellation set of the source. Figures 4.3–4.7 show SER results for 4QPSK, 16-QAM and 64-QAM modulation, each for block lengths of $L = 100$ and 1000 (only $L = 1000$ for the 64-QAM case). Each of these figures show SER results corresponding to the following estimation schemes for the parameters $\boldsymbol{H}$ and $\boldsymbol{\Psi}$: *i*) the parameters are perfectly known at the receiver (asterisk), *ii*) the parameters are estimated using only the initial training; i.e., EM and IGID are turned off (plus) *iii*) the parameters are estimated using the EM algorithm with 10% of the block used for training (x), and *iv*) the parameters are estimated using the IGID algorithm, again with 10% of the block of data used for training (circle). For the large L and small constellation case, it can be seen from the figures that the performance of the IGID algorithm is very close to that of the EM algorithm. However, it is seen that IGID's performance degrades slightly for decreasing values of L and increasing constellation size. For all the simulation scenarios considered in this study, the number of iterations for the IGID algorithm never exceeded five.

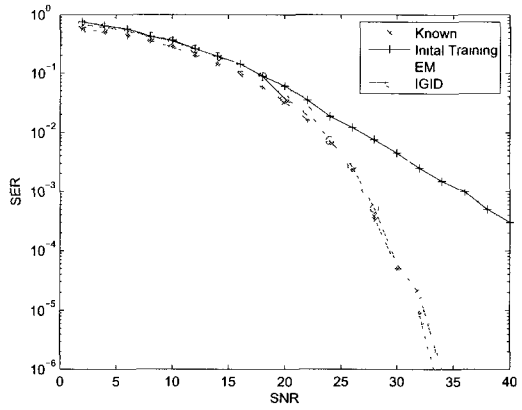


Figure 4.3: Symbol Error Rate (SER) curves for QPSK modulation for a block length of $L = 100$.

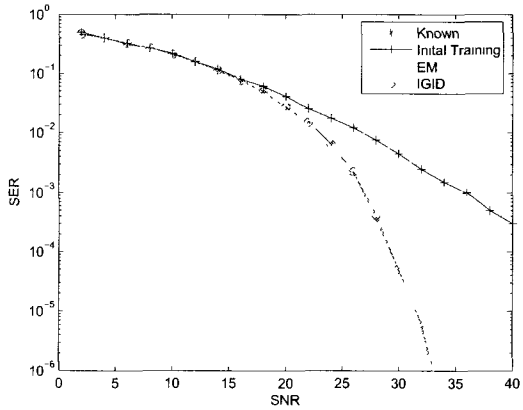


Figure 4.4: Same as Figure 4.3, except the block length $L = 1000$.

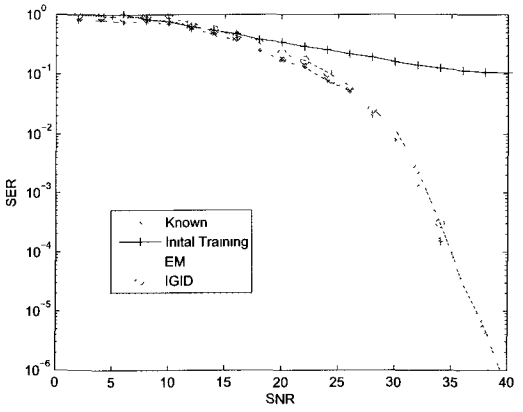


Figure 4.5: SER curves for 16-QAM modulation for a block length of $L = 100$.

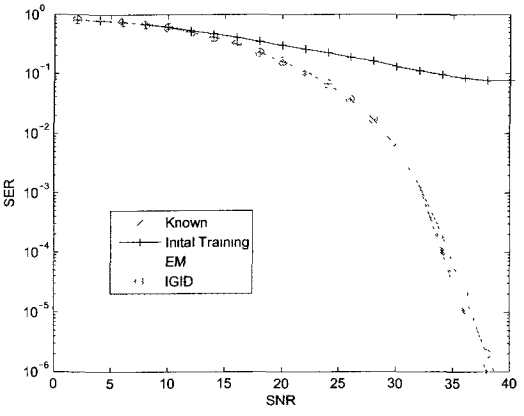


Figure 4.6: SER curves for 16-QAM modulation for a block length of $L = 1000$.

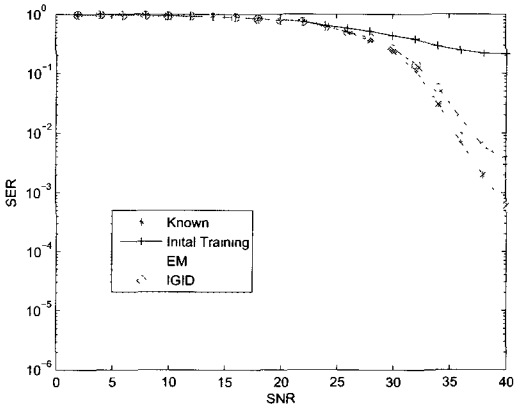


Figure 4.7: SER curves for 64-QAM modulation for a block length of $L = 1000$.

Simulations were performed where the percentage of training symbols relative to the block length was varied. It was determined that the SER performance is insensitive to training lengths above 10% when $L = 100$, and above roughly 5% when $L = 1000$. However, the number of iterations required for convergence changes somewhat with training length.

SER simulations were also conducted for the JADE [21] algorithm, which is deterministic due to the fact it does not exploit the specific distribution of the source nor the parameters. For the same simulation scenarios, it was found that the performance of the JADE method is up to about 3-4 dB worse than the IGID algorithm. This demonstrates the general idea that stochastic ML estimation is usually better than its deterministic counterpart.

In Figure 4.8, we show an example of the convergence of the IGID algorithm, for the same simulation scenario as in Figure 4.1. This figure shows the KL distance between p_t and q_t vs. the iteration index t. As expected, this measure converges monotonically towards zero.

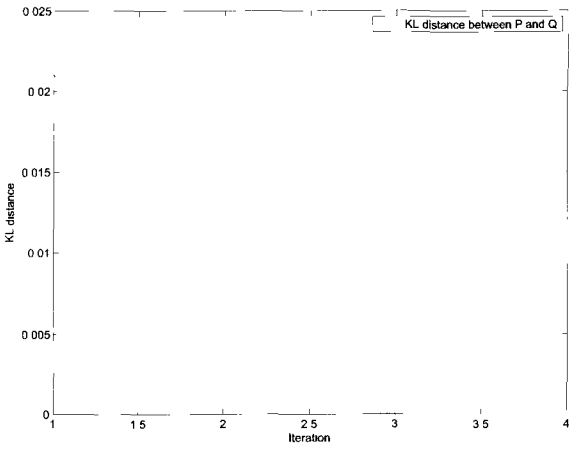


Figure 4.8: $D(p_t||q_t)$ vs. iteration index t.

4.4.3 Discussion

In addition to considering estimation performance, it is important to notice the considerable superiority in terms of computational complexity of the IGID algorithm over the EM-based algorithm. The complexity of the IGID algorithm is dominated by the inversion of the matrix $\boldsymbol{P}_{(M+N)\times(M+N)}$ in (4.27), which is on order of $\mathcal{O}((M+N)^3)$ using the Cholesky factorization [46], where M and N are the number of inputs and outputs, respectively. When the inputs are chosen from a set of discrete values, we see that the complexity of the IGID algorithm is independent of the number of input values and is on the order of $(M+N)^3$. This is in contrast to the complexity of the conventional EM-based algorithm, summarized in Appendix II, which is dominated by the computation of $\overline{\boldsymbol{x}\boldsymbol{x}^T}$ (defined in Appendix E), and is on the order of $\mathcal{O}(LM^2C^M)$ where L, M and C are the data block length, the number of input sources, and the number of discrete points in the constellation. Notice the exponential growth in complexity with C and M. Thus, for spectrally efficient signalling schemes, which use a large value of C, the IGID algorithm can be orders of magnitude faster than the EM algorithm.

For the current scenario using 16-QAM signalling, with $M = N = 2$, $C = 16$,

and $L = 1000$, the complexity of the IGID algorithm is of the order 64, compared to $1,024,000$ for the EM-based algorithm. These figures are $(64; 51,200)$ and $(64; 10,240)$ when the block length reduces to $L = 500$ and $L = 100$ (appropriate for fast-fading channels), respectively. The ratio of the actual FLOP counts for each iteration of the two algorithms as measured by MATLAB© is about $(15,000)$, (800), and (150) for $L = 1000$, $L = 500$, and $L = 100$, respectively. This figures show a noticeable improvement in the execution speed of the proposed algorithm.

We have made the assumption that the source data and the observations are Gaussian distributed. This approximation is not far from reality in the low-SNR regime, due to the prominent effect of the noise on the output distribution. However, in high-SNR regime, since the noise effect is not as significant, the validity of this assumption diminishes. This seems to be the main reason for the discrepancy which exists between the performance of the IGID algorithm relative to that of previous EM algorithms. Nevertheless, this small deviation has a negligible effect on symbol error, as noted in the previous simulation results. However, we have noted that this marginal decrease in performance is accompanied by a significant gain in computational cost.

In this chapter, the IGID algorithm has been developed for the following form of model:

$$\boldsymbol{y}(t) = \boldsymbol{H}\boldsymbol{x}(t) + \boldsymbol{v}(t)$$

where $\boldsymbol{y}(t)$ is a vector of observations, $\boldsymbol{H}$ is a matrix which is a function of unknown parameters, $\boldsymbol{x}(t)$ is the input vector, and $\boldsymbol{v}(t)$ is a noise vector. This model appears in many signal processing problems, such as direction of arrival estimation, tracking, model identification with hidden Markov models, in blind identification of channels, etc. Thus, the proposed method is applicable not only to the blind identification problem, but to many other problems in signal processing as well.

4.5 Conclusions

In this chapter an information geometric approach to blind identification was presented. Based on information geometry, a low-complexity iterative identification procedure, called the *IGID algorithm*, for blind identification of unknown parameters in a multi-input multi-output (MIMO) system with Gaussian distributed noise was proposed. The algorithm is an iterative solution to the *incomplete-data problem* posed by maximum likelihood (ML) estimation of parameters in a linear Gaussian MIMO system when only the output observations are available. The IGID algorithm involves two iterative minimizations, corresponding to projections onto the likelihood PD (*probability distribution*) set and the empirical PD set, respectively. A Gaussian assumption on the source allows us to develop closed–form expressions for the projection operations. The performance of the IGID algorithm in blind identification of the channel gain matrix in a MIMO communication system was investigated. Simulation results showing the symbol-error-rate (SER) behavior are given. It is shown by simulation that the performance of the IGID algorithm is only slightly degraded relative to that of previous EM-based algorithms [5]; however, a noticeable improvement in computational cost is realized.

Chapter 5

An EM Algorithm for State Estimation

5.1 Introduction

In most solutions to state estimation problems, both linear and nonlinear, it is generally assumed that the state transition process and the measurement process parameters are known a priori. For instance, in target tracking using a nonlinear state space model, the extended Kalman filter (EKF) assumes that the process and measurement matrices as well as corresponding noise statistics are known [13]. However, there are situations in which the model parameters are not known a priori or they are known with some degree of uncertainty. In these state estimation problems neither a set of certain training data is available to accurately identify the model uncertainties, nor are accurate models of the measurement process available to precisely estimate the states using conventional estimation procedures. In these circumstances, standard estimation algorithms, which are based on perfect knowledge of the model parameters, are not accurate anymore. To solve this problem, we must perform optimum state estimation in the presence of model uncertainty; i.e., perform model identification

while tracking.

There are three main categories of methods proposed to perform this task. The classical remedy is to treat the unknown parameters as extra state variables and augment the state vector by the unknown parameters [13]. For a review of these methods refer to [3]. See also the reduced state estimator approach [75].

The second category consists of the so-called multiple model (MM) estimators [13]. An adaptive filtering algorithm decides the most appropriate model from a number of different but predefined model dynamics during the estimation process. The generalized pseudo-Bayesian estimator (GSBS) and the interactive multiple model estimation (IMM) procedure are among the best known examples [13] of this type of method. The MM estimators with variable structure are another example of this type of estimator. For a comprehensive review of these methods, interested readers are referred to the survey paper [72] along with [63, 64, 50]. MM algorithms show promising performance in tracking maneuvering targets whose dynamics are predictable. However, their ability to handle model uncertainties is limited to the "model dictionary" available. Furthermore, they often require a long time to acquire track, as shown in the simulation section of this chapter.

The key idea in the third category of algorithms is to divide the problem of state estimation in the presence of the model uncertainty into two joint problems; i.e., state estimation and model identification [67]. First, assuming that the model is known perfectly, the states are estimated. Then the estimated states with their corresponding measurements are used to identify the model parameters. Perhaps the first paper in this regard was [25]. Also, in [3], an optimality test was derived to adjust a Kalman filter when the noise statistics are not known exactly. Later in [69] and [52], joint simultaneous state estimation and model identification for the scalar state estimation case, in the presence of unknown model parameters, was studied. See also [11, 83, 53, 76, 49] for similar approaches.

This third category is the one chosen for this chapter. Interestingly, this approach can be cast in an expectation maximization (EM) context [35]. See [82] in which a general framework for solving the general joint estimation-identification of *linear Gaussian* models was presented. Refer to [97, 98] for a similar application of the EM algorithm for linear state estimation with uncertain model parameters. The main idea in EM-based algorithms is to solve the state estimation problem in the presence of model uncertainty in two iterative steps. In the first step, called the E-step, it is assumed that the model is known perfectly and therefore standard estimation methods are used to estimate the states. Then, in the second step, i.e., the M-step, the estimated states with their corresponding measurements are used to identify the model parameters. Different implementations of the E and the M steps have resulted in different algorithms suitable for different applications.

In this chapter we extend the approaches of [82] and others with regard to the problem of model identification while tracking. Here, we extend the previous work to the case where the measurement model is nonlinear and unknown, subject to the restriction that it can be accurately represented as a mixture of Gaussian (MoG) kernels, and that the Cramer–Rao bound for all the model parameters and the states, given the observations, exists. A specific EM procedure called the EM-PF algorithm is presented.

In the E-step of the proposed algorithm, an approximation of the posterior distribution of the states given the measurements is formulated. This distribution is then used to estimate the states. In nonlinear systems this conditional density is generally non-Gaussian and can be quite complex. We use a particle filter [36] algorithm to estimate and recursively update this posterior distribution in time. Because the EM algorithm is sensitive to initialization, the particle filter is initialized using a Metropolis-Hastings Monte-Carlo Markov Chain (MH-MCMC) [36] procedure. This

greatly assists the algorithm in converging to the global optimum. In the maximization (M) step, the unknown measurement process is approximated by fitting the observations to an MoG model using the current estimate of the states. A closed–form maximum likelihood procedure for determining the parameters of the MoG model is given.

Finally, the proposed EM-PF algorithm is applied to two nonlinear state estimation problems with model uncertainties. First, we consider a typical bearing-only tracking problem where the sensors have an unknown measurement bias. In this example we treat the observation model in the presence of sensor biases as unknown. It is shown that the EM-PF algorithm is capable of successfully estimating the position and velocity states and therefore can accommodate model uncertainty and correct the misalignment caused by the sensor bias. Then, we approach a sensor registration problem in which different sensors with different unknown bias values combine their measurements for state estimation. Here again we treat the observation model as unknown. We show that the sensor registration is performed successfully and the effect of sensor bias is suppressed by the algorithm. Even though in each of the above examples it may be possible to gain better performance by exploiting the known form of the nonlinear model, we demonstrate that the proposed method is applicable to situations where very little is known about the structure of the observation model.

The structure of chapter is as follows. In Section 5.2, the general framework for the EM algorithm is introduced. The details of the proposed EM–PF algorithm follow. The implementation of the E-step using a particle filter, and the M-step by fitting an MoG model to the estimated data, are provided in Section 5.3. Then in Section 5.4, the proposed method is applied to a nonlinear bearing-only tracking problem (similar to the one in [65]) with uncertain model parameters. Also, Section 5.5 presents the application of the EM-PF algorithm to a sensor-registration problem in a multisensor tracking scenario. Simulation results are presented for each application.

Throughout the text, the notation $\mathcal{N}(\boldsymbol{m}, \boldsymbol{\Sigma})$ indicates a Gaussian distribution with mean $\boldsymbol{m}$ and covariance $\boldsymbol{\Sigma}$. An upper–case bold symbol (e.g., $\boldsymbol{A}$) denotes a matrix, and a lower–case bold symbol denotes a vector. If the vector is a function of time, e.g., $\boldsymbol{z}(t)$, then the corresponding symbol without the time index (e.g., $\boldsymbol{z}$) denotes the set of all values of the vector over the range of the temporal index; e.g., $\boldsymbol{z}$ denotes $\{\boldsymbol{z}(t)|t = 1, \ldots, L\}$, where L is the number of data points.

Throughout, t where $t = 1, \ldots, L$ denotes the *discrete* time index, $k = 1, 2 \ldots$ is the EM iteration index where $k = 1$ is the initialization step, and i, where $i = 1, \ldots, N$ is the particle index, where N is the number of particles used in the particle filter.

5.2 Nonlinear State Estimation using EM

State estimation in a nonlinear state-space dynamical system whose evolution process is described as

$$\boldsymbol{x}(t+1) = \boldsymbol{f}(\boldsymbol{x}(t)) + \boldsymbol{u}(t), \tag{5.1}$$

consists of estimating the state data vector $\boldsymbol{x}$ using a sequence of noisy measurements given by the following model:

$$\boldsymbol{z}(t) = \boldsymbol{h}\big(\boldsymbol{x}(t), \boldsymbol{\theta}\big) + \boldsymbol{v}(t), \qquad t = 1, 2, \ldots, \tag{5.2}$$

where t is the *discrete* time index, $\boldsymbol{x}(t) \in \mathbb{C}^M$ and $\boldsymbol{z}(t) \in \mathbb{C}^J$ are the state variable and the noisy output measurement vectors respectively, and $\boldsymbol{u}(t) \in \mathbb{C}^M$ is assumed to be an *i.i.d* noise processes, whose probability density function is assumed known and possibly non–Gaussian. The vector $\boldsymbol{v}(t) \in \mathbb{C}^J$ is a zero-mean Gaussian noise variable with unknown covariance $\boldsymbol{Q}$. The noise $\boldsymbol{v}(t)$ is assumed uncorrelated in time; i.e., $E\big(\boldsymbol{v}(t_1)\boldsymbol{v}(t_2)\big) = \delta_{t_1,t_2}\boldsymbol{Q}$.

Also, the vector valued functions $\boldsymbol{f}$, $\boldsymbol{f} : \mathbb{C}^M \mapsto \mathbb{C}^M$, and $\boldsymbol{h}$, $\boldsymbol{h} : \mathbb{C}^M \mapsto \mathbb{C}^J$ are assumed to be smooth but otherwise are arbitrary. We assume that the function $\boldsymbol{f}(\cdot)$

is known, whereas uncertainty may exist in the observation model $\boldsymbol{h}(\cdot)$.

A major focus of this chapter is how to model the partially known or unknown function $\boldsymbol{h}(\cdot)$. If a model which takes into account any known structure in the measurement process is available, then that model should be used in the proposed method. Any uncertainty is expressed in a parameter vector $\boldsymbol{\theta}$. On the other hand, it is also possible to assume no structure on $\boldsymbol{h}(\cdot)$, as is done with our examples in Sects. 5.4 and 5.5. We model this function as an MoG, again parameterized by the vector $\boldsymbol{\theta}$, in a manner to be described later in Sect. 5.3.2.

A restriction on the proposed methodology is that the Cramer–Rao bound on all the states and on all the parameters $\boldsymbol{\theta}$ which describe the model given the observations, must exist. Here we do not discuss conditions for which the bound exists. However, it is clear that the proposed formulation will place restrictions on the class of problems that may be considered.

When the model is known completely, maximum likelihood (ML) state estimation results in a filtering problem, which can be solved using, e.g., the EKF, the particle filter, or the unscented Kalman filter [56]. Also, in the case where when the model structure is known but contains a number of unknown parameters, and a training set consisting of corresponding state and measurement data is available, then the states and unknown model parameters can be jointly estimated using maximum likelihood (ML) procedures, as is common practice in communication systems. However, in the case considered here where we assume that no training set is available, and the measurement function $\boldsymbol{h}$ is uncertain or unknown, standard estimation algorithms that assume perfect knowledge of the model parameters are not accurate. In this case it is desirable to jointly estimate the state vectors and the observation model using an EM technique which blindly incorporates model uncertainty, as is proposed in this chapter.

To estimate the states in the presence of model uncertainty, we use the *variational*

form of the EM algorithm [82]. The log likelihood of observations is defined as:

$$\mathcal{L}(\boldsymbol{\theta}) = \log p(\boldsymbol{z}|\boldsymbol{\theta}) = \log \int_{\chi} p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) d\boldsymbol{x} \tag{5.3}$$

where χ is the range of the state variables, $\boldsymbol{z} = [\boldsymbol{z}^T(1), ..., \boldsymbol{z}^T(L)]^T \in \mathbb{C}^{JL}$ is the entire sequence of observed measurements, $\boldsymbol{x} = [\boldsymbol{x}^T(1), ..., \boldsymbol{x}^T(L)]^T \in \mathbb{C}^{ML}$ are all the state variables, $\boldsymbol{\theta}$ is the vector of parameters describing the MoG model, and L is the number of observation points.

Maximizing this function can often be intractable in the nonlinear/non-Gaussian case. Therefore, an alternative procedure is to define a variational distribution $U(\boldsymbol{x})$ over the hidden state variables, that allows us to obtain a lower bound on the expected likelihood [73, 82]:

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}) &= \log \int_{\chi} p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) d\boldsymbol{x} \\
&= \log \int_{\chi} U(\boldsymbol{x}) \frac{\boldsymbol{p}(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta})}{U(\boldsymbol{x})} d\boldsymbol{x} \\
&\geq \int_{\chi} U(\boldsymbol{x}) \log \frac{p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta})}{U(\boldsymbol{x})} d\boldsymbol{x} \qquad (5.4) \\
&= \int_{\chi} U(\boldsymbol{x}) \log p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) d\boldsymbol{x} - \int_{\chi} U(\boldsymbol{x}) \log U(\boldsymbol{x}) d\boldsymbol{x} \\
&= \int_{\chi} U(\boldsymbol{x}) \log p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) d\boldsymbol{x} + H(U) \qquad (5.5) \\
&\triangleq \mathcal{F}(U, \boldsymbol{\theta}),
\end{aligned}$$

where (5.4) follows from Jensen's inequality [13] and $H(U)$ in (5.5) is the entropy of the distribution U. It is straightforward to show that the equality in (5.4) is satisfied for $U^{\star}(x) = p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\theta})$.

The EM algorithm alternates between maximizing $\mathcal{F}$ with respect to the distribution $U(\boldsymbol{x})$ and the parameters $\boldsymbol{\theta}$, respectively. Starting from some initial parameters $\boldsymbol{\theta}_0$ the algorithm iteratively applies

$$E - Step: \quad U_{k+1} = \arg\max_{U} \mathcal{F}(U_k, \boldsymbol{\theta}_k) \tag{5.6}$$

$$M - Step: \quad \boldsymbol{\theta}_{k+1} = \arg\max_{\boldsymbol{\theta}} \mathcal{F}(U_{k+1}, \boldsymbol{\theta}_k), \tag{5.7}$$

where k is the EM iteration index. The primary purpose of the the E-step is to estimate the hidden states. This is accomplished by determining the best distribution $U^\star = p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\theta})$ which makes the expectation of log-likelihood maximum. A conditional–mean estimate of the states is then readily available from this distribution. The M-step involves estimating the model parameters $\boldsymbol{\theta}$ using the states estimated in the previous E-step and their corresponding measurements. Since, at the end of each E-step the likelihood function $\mathcal{F}$ meets the equality for $U^\star(x)$, then $\mathcal{F}(U^\star_{k+1}, \boldsymbol{\theta}_k) = \mathcal{L}(\boldsymbol{\theta}_k)$. Also, because in the M-step the optimization is over $\boldsymbol{\theta}$, it is guaranteed that the likelihood will not decrease in any iteration.

5.3 The EM–PF Algorithm

The overall operation of the proposed EM–PF algorithm for estimating states in the presence of model uncertainties, nonlinear models and non–Gaussian noise is shown in Figure 5.1. The algorithm as shown in this figure operates in batch mode, using a finite set of observations $\boldsymbol{z}(t), t = 1, \ldots, L$. Since we wish to estimate the states $\boldsymbol{x}_k(t)$ over this same interval, the problem may be cast as a fixed interval smoothing problem. It is assumed that the parameters $\boldsymbol{\theta}_k$ describing the model do not change significantly over this interval. In the situation of interest here, where the observation noise is non–Gaussian or the model is nonlinear, the distribution $p(\boldsymbol{x}_k(t)|\boldsymbol{z}, \boldsymbol{\theta}_k)$, which is critical to the E–step, cannot be evaluated analytically. In the proposed EM-PF algorithm, this distribution is approximated using a particle filter.

5.3.1 The E-step: Estimation of States by the Particle Filter

At the kth iteration of the EM algorithm, the distribution of interest for the E–step is $p(\boldsymbol{x}_k(t)|\boldsymbol{z}, \boldsymbol{\theta}_k)$, for $t = 1, \ldots, L$. When the noise is non–Gaussian or the

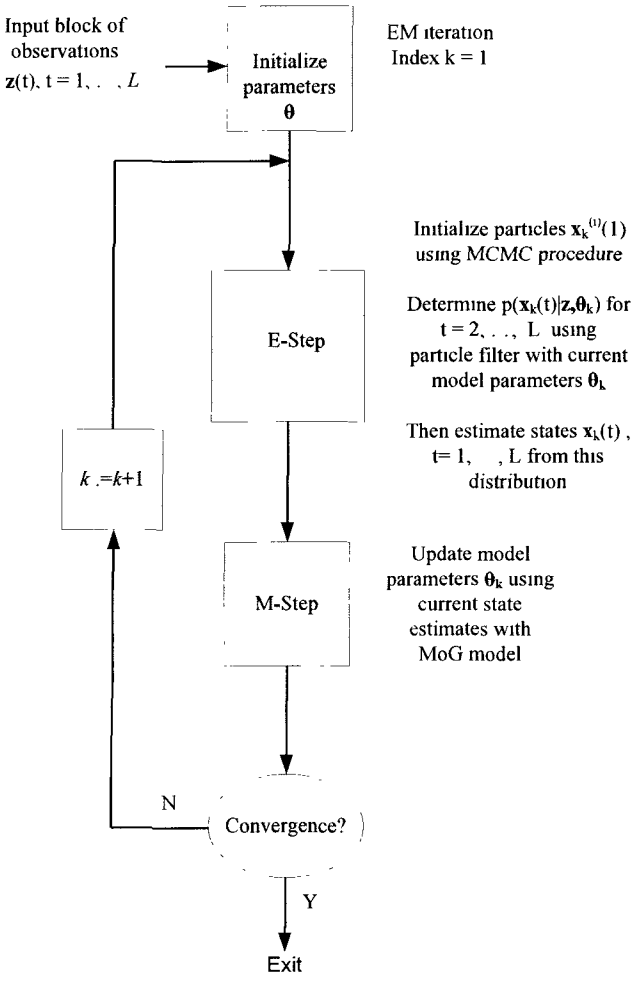


Figure 5.1: Block diagram of the EM–PF algorithm, which gives state and model parameter estimates, over the block $t = 1, \ldots, L$.

model is nonlinear, this distribution can be intractable. Instead, an approximation $\hat{p}_N(\boldsymbol{x}_k(t)|\boldsymbol{z}, \boldsymbol{\theta}_k)$ to this distribution is used, which is propagated in time by a particle filter. At the beginning of the kth E-step, it is assumed that the parameter vector $\boldsymbol{\theta}_k$ has been estimated within the previous M-step and therefore is known.

Given $\hat{p}_N(\boldsymbol{x}_k(t)|\boldsymbol{z}, \boldsymbol{\theta}_k)$, the states $\boldsymbol{x}_k(t)$ can be estimated as, e.g., the conditional mean of this distribution at any time $t = 1, \ldots, L$. Then in the M-step, the estimated states and their corresponding measurements are used to identify the measurement function $h()$, parameterized by $\boldsymbol{\theta}_k$. This vector is estimated using the state-measurement pairs estimated in the E-step. The E– and M–steps iterate until convergence.

The presentation on particle filters here is necessarily brief; readers are referred to [10, 36] for further background. We first explain the case for the *filtering* distribution; i.e., approximation of the filtering distribution $p(\boldsymbol{x}_k(t)|\boldsymbol{z}_{1:t}, \boldsymbol{\theta}_k)$[1]. We later extend the treatment to the *fixed–interval smoothing* problem, i.e., approximation of the distribution $p(\boldsymbol{x}_k(t)|\boldsymbol{z}_{1:L}, \boldsymbol{\theta}_k)$, for any $t \in [1, \ldots, L]$, which is the problem of relevance here.

The quantity $\hat{p}_N(\boldsymbol{x}_k(t)|\boldsymbol{z}_{1:t}, \boldsymbol{\theta}_k)$ is specified by a set $\{\boldsymbol{x}_k^{(i)}(t), w_k^{(i)}(t)\}_{i=1:N}$, where the $\boldsymbol{x}_k^{(i)}(t)$ are samples (particles) of the states, that are used to compose the desired distribution. The quantity N is the number of particles, and $w_k^{(i)}(t)$ are the respective filtering weights, whose calculation is described below. The approximation $\hat{p}_N(\boldsymbol{x}_k(t)|\boldsymbol{z}_{1:t}, \boldsymbol{\theta}_k)$ is given by

$$\hat{p}_N(\boldsymbol{x}_k(t)|\boldsymbol{z}_{1:t}, \boldsymbol{\theta}_k) = \sum_{i=1}^{N} w_k^{(i)}(t)\delta\big(\boldsymbol{x}_k(t) - \boldsymbol{x}_k^{(i)}(t)\big), \tag{5.8}$$

where $\delta(\cdot)$ is the Dirac delta function.

The unnormalized weights at time t can be recursively updated from those at time

[1]The notation $\boldsymbol{y}_{a:b}$ is commonly used in the particle filtering literature and implies all values of $\boldsymbol{y}$ from time a to b.

$t-1$ at EM iteration k by [10, 36]

$$\tilde{w}_k^{(i)}(t) = w_k^{(i)}(t-1)\frac{p(\boldsymbol{z}(t)|\boldsymbol{x}_k^{(i)}(t),\boldsymbol{\theta}_k)p(\boldsymbol{x}_k^{(i)}(t)|\boldsymbol{x}_k^{(i)}(t-1))}{r(\boldsymbol{x}_k^{(i)}(t)|\boldsymbol{x}_k^{(i)}(t-1),\boldsymbol{z}(t))}, \qquad i=1,\ldots,N. \quad (5.9)$$

The normalized weights $w_k^{(i)}(t)$ are then calculated as

$$w_k^{(i)}(t) = \frac{\tilde{w}_k^{(i)}(t)}{\sum_{i=1}^{N}\tilde{w}_k^{(i)}(t)}, \qquad i=1,\ldots,N. \quad (5.10)$$

The quantities $\boldsymbol{x}_k^{(i)}(t)$ in (5.9) are the particles, which are samples drawn from a proposal distribution $r\big(\boldsymbol{x}_k^{(i)}(t)\big|\boldsymbol{x}_k^{(i)}(t-1)\big)$. This distribution is chosen to be easy to sample from, and to resemble the desired distribution $p(\boldsymbol{x}_k(t)\big|\boldsymbol{z},\boldsymbol{\theta}_k)$ as closely as possible. We choose the proposal distribution to be a normal distribution:

$$r(\boldsymbol{x}_k^{(i)}(t)|\boldsymbol{x}_k^{(i)}(t-1)) \sim \mathcal{N}(\boldsymbol{x}_k(t-1),\sigma_r^2), \quad (5.11)$$

where σ_r^2 is chosen such that the support of the distribution properly covers the current state $\boldsymbol{x}_k(t)$.

The distribution $p(\boldsymbol{z}(t)|\boldsymbol{x}_k^{(i)}(t),\boldsymbol{\theta}_k)$ in (5.9) is the likelihood, and is determined from (5.2), given $\boldsymbol{\theta}_k$ and knowledge of the distribution of $\boldsymbol{v}$. The distribution $p(\boldsymbol{x}_k^{(i)}(t)|\boldsymbol{x}_k^{(i)}(t-1))$ in (5.9) is the prior distribution on the states and is given from (5.1), knowing the distribution of $\boldsymbol{u}(t)$. Thus, the method propagates the desired distribution $p(\boldsymbol{x}_k(t)\big|\boldsymbol{z}_{1:t},\boldsymbol{\theta}_k)$ in time at each value of t by first, drawing particles $\boldsymbol{x}_k^{(i)}(t), i=1,\ldots,N$ from the proposal distribution $r(\cdot|\cdot)$. Then, using the particles and the observations, the respective distributions in (5.9) can be evaluated. The weights $w_k^{(i)}(t)$ are then updated using (5.9) and (5.10), whereupon the desired approximate distribution is obtained by (5.8).

We now extend this treatment to the fixed–interval smoothing problem. It is shown in [37] that the smoothing distribution is given as

$$\hat{p}_N(\boldsymbol{x}_k(t)\big|\boldsymbol{z}_{1\cdot L},\boldsymbol{\theta}_k) = \sum_{i=1}^{N} w_k(t|L)^{(i)}\delta(\boldsymbol{x}_k(t)-\boldsymbol{x}_k^{(i)}(t)) \quad (5.12)$$

for any $t \in [1, \ldots, L]$. Thus, only the weights change in going from the filtering to the smoothing problem. The smoothing weights $w_k^{(i)}(t|L)$ are calculated according to the following algorithm [37]:

1. Initialization at time $t = L$:
   - For $i = 1, \ldots, N$, $w_k^{(i)}(L|L) = w_k^{(i)}(L)$.
2. for $t = L, \ldots, 1$
   - for $i = 1, \ldots, N$, evaluate the smoothing weights:

$$w_k^{(i)}(t|L) = \sum_{m=1}^{N} w_k^{(m)}(t+1|L) \frac{w_k^{(i)}(t) p(\boldsymbol{x}^{(m)}(t+1) \big| \boldsymbol{x}^{(i)}(t))}{\left[\sum_{\ell=1}^{N} w_k^{(\ell)}(t) p\big(\boldsymbol{x}^{(m)}(t+1) \big| \boldsymbol{x}^{(\ell)}(t)\big)\right]}. \quad (5.13)$$

Using (5.12), conditional mean state estimates $\hat{\boldsymbol{x}}_k(t)$ can be obtained for any time $t = 1, \ldots, L$ by

$$\begin{aligned} \hat{\boldsymbol{x}}_k(t) &= \int_{\mathcal{X}} \boldsymbol{x}_k(t) p(\boldsymbol{x}_k(t) \big| \boldsymbol{z}_{1:L}, \boldsymbol{\theta}_k) d\boldsymbol{x}_k(t) \\ &\approx \int_{\mathcal{X}} \boldsymbol{x}_k(t) \hat{p}_N(\boldsymbol{x}_k(t) \big| \boldsymbol{z}_{1:L}, \boldsymbol{\theta}_k) d\boldsymbol{x}_k(t) \\ &= \sum_{i=1}^{N} w_k^{(i)}(t|L) \boldsymbol{x}_k^{(i)}(t). \end{aligned} \quad (5.14)$$

In the sequel, for ease of notation we write $\boldsymbol{z}$, implying $\boldsymbol{z}_{1:L}$

The problem with the particle filter is that after a few time steps, all but a very few of the particles have negligible weights. This degeneracy problem results in inefficient use of the particles. There are a number of proposed resampling techniques that correct this problem. A simple minimum variance scheme first proposed by Kitagawa [59], and applied to a tracking problem [65], is used in this chapter. This re-sampling technique probabilistically replicates particles with large weights and discards particles with small weights, so that our set of particles better represents the required distribution.

Initialization of the particle filter at $t = 1$: The initial particles at time $t = 1$ for each EM iteration k must be chosen carefully, otherwise the particle filter may lose track later in time. For this purpose, we consider the Metropolis-Hastings (MH) algorithm, which is a Monte Carlo Markov chain (MCMC) procedure, for generating samples from the initial posterior distribution $\pi = p(\boldsymbol{x}_k(1)|\boldsymbol{z}(1), \boldsymbol{\theta}_k)$. Ideally, we would like to use the exact distribution $p(\boldsymbol{x}_k(1)|\boldsymbol{z}, \boldsymbol{\theta}_k)$; however, this is not possible for reasons of tractability, so we use π as an approximation. As described below, the MCMC process is iterative; each iteration places an underlying Markov chain in a different state, which corresponds to a sample; thus, a potential candidate sample is drawn in each iteration. An appropriate number of initial iterations (referred to as the *burn–in* period), are required before the underlying Markov chain establishes equilibrium. Only after equilibrium is established are the samples distributed according to the desired distribution π; therefore, the burn–in samples are discarded. After the burn-in period completes, N useful samples are drawn by executing N additional iterations. These additional samples serve as the initial particles $x_k^{(i)}(1)$ for the particle filter. Since these initial particles are already distributed according to the approximate desired posterior distribution, the corresponding weights are all initialized to unity.

By choosing a proposal density $q(\boldsymbol{x}|\cdot)$ which may be different from $r(\cdot|\cdot)$, the following procedure generates samples, $\boldsymbol{x}_k^{(i)}(1)$ from $\pi \triangleq p(\boldsymbol{x}_k(1)|\boldsymbol{z}(1), \boldsymbol{\theta}_k)$:

for $i = 1, \ldots N$, after equilibrium of π is reached:

1) Sample $\boldsymbol{x}^* \sim q(\boldsymbol{x}|\boldsymbol{x}^{(i-1)})$

2) Evaluate

$$\alpha\left(\boldsymbol{x}^{(i-1)}, \boldsymbol{x}^*\right) \triangleq \min\left\{1, \frac{\pi(\boldsymbol{x}^*)q(\boldsymbol{x}^{(i-1)}|\boldsymbol{x}^*)}{\pi(\boldsymbol{x}^{(i-1)})q(\boldsymbol{x}^*|\boldsymbol{x}^{(i-1)})}\right\} \tag{5.15}$$

3) assign $\boldsymbol{x}_k^{(i)}(1) = \boldsymbol{x}^*$ with probability $\alpha(\boldsymbol{x}^{(i-1)}, \boldsymbol{x}^*)$.

We choose the proposal density $q(\boldsymbol{x}|\boldsymbol{x}^{(\iota-1)})$ to be an easy-to-sample distribution; e.g., the Gaussian distribution:

$$q(\boldsymbol{x}|\boldsymbol{x}^{(\iota-1)}) \sim \mathcal{N}(\boldsymbol{x}^{(\iota-1)}, \sigma_q^2), \tag{5.16}$$

where, in this study, the variance σ_q^2 is chosen empirically, so that the support region of the proposal density properly covers the state space around the initial state $\boldsymbol{x}_k(1)$.

The distribution π for evaluating the samples in (5.15) can be obtained by Bayes' rule, assuming that $\boldsymbol{\theta}_k$ is independent of $\boldsymbol{x}_k(t)$, as follows:

$$\pi = \frac{p(\boldsymbol{z}(1)|\boldsymbol{x}_k(1), \boldsymbol{\theta}_k))p(\boldsymbol{x}_k(1)|\boldsymbol{\theta}_k)}{p(\boldsymbol{z}(1)|\boldsymbol{\theta}_k)} \tag{5.17}$$

where the likelihood distribution $p(\boldsymbol{z}(1)|\boldsymbol{x}_k(1), \boldsymbol{\theta}_k)$ is obtained from the measurement process (5.2) and the prior distribution $p(\boldsymbol{x}_k(1)|\boldsymbol{\theta}_k) = p(\boldsymbol{x}_k(1))$ is assumed to be uniform. The value of $p(\boldsymbol{z}(1)|\boldsymbol{\theta}_k)$ is irrelevant for the purposes at hand, since it is independent of $\boldsymbol{x}$ and hence cancels in (5.15).

5.3.2 The M-Step

In this section we demonstrate the M–step for the general case where no structure is available for $\boldsymbol{h}(\cdot)$. Here, $\boldsymbol{h}(\cdot)$ is modelled as a mixture of Gaussian kernels. If specific structure is available, then the procedure can be modified accordingly.

By substituting the posterior distribution of the states given the observations obtained in the E-step ($U_{k+1}^{\star} = p(\boldsymbol{x}_k|\boldsymbol{z}, \boldsymbol{\theta}_k)$) into (5.5), the required optimization for the M-step of the kth EM iteration becomes

$$\boldsymbol{\theta}_{k+1} = \arg\max_{\boldsymbol{\theta}} \int_{\chi} p(\boldsymbol{x}_k|\boldsymbol{z}, \boldsymbol{\theta}_k) \log p(\boldsymbol{x}_k, \boldsymbol{z}|\boldsymbol{\theta}) d\boldsymbol{x}_k(1) \ldots d\boldsymbol{x}_k(L). \tag{5.18}$$

To proceed with this optimization, we incorporate our model for the observation function $\boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{\theta})$ in (5.2). If some structure regarding $\boldsymbol{h}(\cdot)$ is available, then it

should be incorporated into a suitable model. However in our case, no structure on $\boldsymbol{h}(\cdot)$ is assumed, and it is modeled as a mixture of Gaussians with P components, as

$$h\big(\boldsymbol{x}(t),\boldsymbol{\theta}\big) \approx \sum_{p=1}^{P} \boldsymbol{m}_p g_p\big(\boldsymbol{x}(t)\big) + \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{b} \tag{5.19}$$

where the parameters $\boldsymbol{m}_p \in \mathbb{C}^J$ are the coefficients of the scalar Gaussian kernels g_p, with fixed centers $\boldsymbol{c}_p \in \mathbb{C}^M$ and fixed covariance matrices $\boldsymbol{S}_p \in \mathbb{C}^{M\times M}$. The centers are distributed uniformly over the range of $\boldsymbol{x}$, and the covariances $\boldsymbol{S}_p$ may be assigned arbitrarily.[2] The Gaussian kernels are defined as:

$$g_p(\boldsymbol{x}) = (2\pi)^{M/2}|\boldsymbol{S}_p|^{-\frac{1}{2}} \exp[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{c}_p)^T\boldsymbol{S}_p^{-1}(\boldsymbol{x}-\boldsymbol{c}_p)]. \tag{5.20}$$

The quantity $\boldsymbol{A} \in \mathbb{C}^{J\times M}$ is a constant matrix, $\boldsymbol{b} \in \mathbb{C}^J$ is a constant bias term. The vector $\boldsymbol{\theta} \in \mathbb{C}^{J\times(P+M+1)}$ is therefore defined as

$$\boldsymbol{\theta} \triangleq [\boldsymbol{m}_1, \boldsymbol{m}_2, ..., \boldsymbol{m}_P, \boldsymbol{A}, \boldsymbol{b}], \tag{5.21}$$

which according to the assumptions, is time–invariant. We also define the vector $\boldsymbol{\Phi}(t) \in \mathbb{C}^{(P+M+1)\times 1}$, which includes the time–varying parameters in $\boldsymbol{h}(\cdot)$ as

$$\boldsymbol{\Phi}(t) \triangleq [g_1(\boldsymbol{x}(t)), g_2(\boldsymbol{x}(t)), ..., g_P(\boldsymbol{x}(t)), \boldsymbol{x}(t)^T, 1]^T. \tag{5.22}$$

Then, (5.19) can be written in the form

$$\boldsymbol{h}\big(\boldsymbol{x}(t),\boldsymbol{\theta}\big) \approx \boldsymbol{\theta}\boldsymbol{\Phi}(t). \tag{5.23}$$

We now evaluate the probability distribution $p(\boldsymbol{x}_k, \boldsymbol{z}|\boldsymbol{\theta}_k)$ at EM iteration k in (5.18). This may be evaluated according to

$$\begin{aligned} p(\boldsymbol{x}_k, \boldsymbol{z}|\boldsymbol{\theta}_k) &= p(\boldsymbol{z}|\boldsymbol{x}_k, \boldsymbol{\theta}_k)\; p(\boldsymbol{x}_k|\boldsymbol{\theta}_k) \\ &\propto p(\boldsymbol{z}|\boldsymbol{x}_k, \boldsymbol{\theta}_k) \end{aligned} \tag{5.24}$$

[2] In the following simulations, they were all assigned to the identity matrix.

where the second line follows because the prior distribution of the states is assumed to be independent of the unknown parameters, and is assigned a uniform distribution. The distribution $p(\boldsymbol{z}|\boldsymbol{x}_k, \boldsymbol{\theta}_k)$ is easily obtained as the likelihood distribution obtained using the observation equation (5.2).

The log-likelihood $\log p(\boldsymbol{z}|\boldsymbol{x}_k, \boldsymbol{\theta}_k)$ of a single fully observed data point $\boldsymbol{z}(t)$ under the model at EM iteration k, using (5.23) and (5.24) is then given as

$$-\big[\boldsymbol{z}(t) - \boldsymbol{\theta}_k \boldsymbol{\Phi}_k(t)\big]^H \boldsymbol{Q}_k^{-1} \big[\boldsymbol{z}(t) - \boldsymbol{\theta}_k \boldsymbol{\Phi}_k(t)\big] - \ln|\boldsymbol{Q}| + \text{constant}. \tag{5.25}$$

By substituting the model log-likelihood into (5.18), and combining the terms for $t = 1, \ldots, L$, the relevant M-step optimization is then

$$\min_{\boldsymbol{\theta}, \boldsymbol{Q}} \left\{ \int_{\mathcal{X}} \sum_{t=1}^{L} p(\boldsymbol{x}_k(t)|\boldsymbol{z}, \boldsymbol{\theta}_k) \big[\boldsymbol{z}(t) - \boldsymbol{\theta}\boldsymbol{\Phi}_k(t)\big]^H \boldsymbol{Q}^{-1} \big[\boldsymbol{z}(t) - \boldsymbol{\theta}\boldsymbol{\Phi}_k(t)\big] d\boldsymbol{x} + \ln|\boldsymbol{Q}| \right\}. \tag{5.26}$$

By denoting the expectation over the posterior distribution $p(\boldsymbol{x}_k(t)|\boldsymbol{z}, \boldsymbol{\theta}_k)$ by $\langle\rangle$, the objective function then becomes

$$\min_{\boldsymbol{\theta}, \boldsymbol{Q}} \sum_{t=1}^{L} \Big\langle [\boldsymbol{z}(t) - \boldsymbol{\theta}\boldsymbol{\Phi}_k(t)]^H \boldsymbol{Q}^{-1} [\boldsymbol{z}(t) - \boldsymbol{\theta}\boldsymbol{\Phi}_k(t)] \Big\rangle + \ln|\boldsymbol{Q}_k|. \tag{5.27}$$

It is shown in the Appendix that the solution to the above is given by

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_{k+1} = \left\langle \sum_{t=1}^{L} \boldsymbol{z}(t) \boldsymbol{\Phi}_k^H(t) \right\rangle \left\langle \sum_{t=1}^{L} \boldsymbol{\Phi}_k(t) \boldsymbol{\Phi}_k(t)^H \right\rangle^{-1} \tag{5.28}$$

$$\hat{\boldsymbol{Q}} = \boldsymbol{Q}_{k+1} = \sum_{t=1}^{L} \Big\langle \boldsymbol{z}(t) \boldsymbol{z}^H(t) \Big\rangle - \sum_{t=1}^{L} \Big\langle \hat{\boldsymbol{\theta}} \boldsymbol{\Phi}_k(t) \boldsymbol{z}^H(t) \Big\rangle \tag{5.29}$$

Thus, given the expectations in the angular brackets, the optimal parameters can be obtained by solving a set of linear equations.

The expectations above are evaluated using the particle filter. Given the computed particles $\boldsymbol{x}_k^{(i)}(t)$ and using the approximation (5.8) for the posterior distribution, the

expectation of any function $f(\boldsymbol{x})$ can be approximated in a manner similar to that of (5.14) by

$$\langle f(\boldsymbol{x})\rangle = \int_{\mathcal{X}} f(\boldsymbol{x}_k(t))p(\boldsymbol{x}_k(t)|\boldsymbol{z},\boldsymbol{\theta}_k)d\boldsymbol{x}_k(t) \approx \sum_{i=1}^{N} w_k^{(i)}(t|L)f(\boldsymbol{x}_k^{(i)}(t)). \tag{5.30}$$

The number P of Gaussian kernels is chosen empirically, so that the kernels are positioned sufficiently densely over the region of support of the state variables.

Initialization of the EM Algorithm: The initial parameters $\boldsymbol{\theta}_1$ must be chosen with some care, otherwise the EM algorithm may not converge. In the experiments described in the following sections, successful results were obtained by assigning the Gaussian kernel coefficients $\boldsymbol{m}_p$ to equal values, the kernel centers $\boldsymbol{c}_p$ to a uniformly–spaced grid, and the kernel covariance matrices $\boldsymbol{S}_p$ to the identity. (The parameters $\boldsymbol{c}_p$ and $\boldsymbol{S}_p$ are held fixed throughout the EM iterations.) The matrix $\boldsymbol{A}$ is also assigned to be an identity (padded appropriately with zeros), and the bias $\boldsymbol{b}$ to zero. Given this initial $\boldsymbol{\theta}$, an E–step was performed to obtain the initial states, $\boldsymbol{x}_0(t), t = 1,\ldots,L$.

5.3.3 Summary

Here we give a step–by–step overview of the proposed EM–PF algorithm:

- **Initialization:** $(k = 1)$ Given a set of measurements, $\boldsymbol{z} = \{\boldsymbol{z}(t), t = 1,\ldots,L\}$, initialize the parameter vector $\boldsymbol{\theta}_1$ to suitable values, as described above. Perform an initial E–step to obtain $\boldsymbol{x}_0(t), t = 1,\ldots,L$.

- **Iterate the EM Algorithm: for** $k = 2, 3, \ldots$

  - **E–step:** In the E-step, we estimate the states $\boldsymbol{x}_k(t)$ using the particle filtering approximation $\hat{p}_N(\boldsymbol{x}_k(t)\big|\boldsymbol{z},\boldsymbol{\theta}_k)$ to the posterior distribution $p(\boldsymbol{x}_k(t)\big|\boldsymbol{z},\boldsymbol{\theta}_k)$ with the most current model. More detail is given as follows:

* Initialize the posterior distribution $p(\boldsymbol{x}_k(1)|\boldsymbol{z}(1), \boldsymbol{\theta}_k)$ at the current EM iteration k using the Metropolis–Hastings MCMC method, described in Sect. 5.3.1. The required proposal density $q(\cdot|\cdot)$ for the MH-MCMC algorithm is chosen to be a Gaussian distribution with mean equal to the previous state $\boldsymbol{x}^{(i-1)}$ and variance chosen so that the support of the function covers adequate space around the current state. Set the filtering weights $w_k^{(i)}(t) = 1, i = 1, \ldots, N$.
* propagate $p(\boldsymbol{x}_k(t)|\boldsymbol{z}, \boldsymbol{\theta}_k)$ for $t = 2, 3, \ldots, L$ using the particle filter. The approximate posterior distribution $\hat{p}_N(\boldsymbol{x}_k(t)|\boldsymbol{z}, \boldsymbol{\theta}_k)$ for the fixed–interval smoothing case is given by (5.12) as a function of the smoothing weights $w_k^{(i)}(t|L)$. These weights are propagated to the next time step by the following procedure: first the *filtering* weights $w^{(i)}(t-1)$ are propagated to time t using (5.9) and (5.10). Then, the filtering weights are converted to the smoothing weights $w_k^{(i)}(t|L)$ using the algorithm surrounding (5.13).
* The likelihood $p(\boldsymbol{z}(t)|\boldsymbol{x}_k(t), \boldsymbol{\theta}_k)$ used in (5.9) is given from the MoG model (5.19), knowing the statistics of $\boldsymbol{n}$. The form of prior distribution $p(\boldsymbol{x}(t+1)|\boldsymbol{x}(t))$ also used in (5.9) depends on the underlying physics of the model, as determined by (5.2). Examples are given in Sects. 5.4 and 5.5.
* Once the smoothing weights $w_k^{(i)}(t|L)$ are available, an approximate conditional mean estimate of the states $\boldsymbol{x}_k(t)$ is given at each time through (5.14).

– **M–step**: The approximated states with their corresponding measurements are then used in the M-step to re-estimate the parameters of the MoG, i.e., the parameter vector $\boldsymbol{\theta}_{k+1}$ and the model noise covariance $\boldsymbol{Q}_{k+1}$ using (5.28) and (5.29) respectively. These estimated parameters are used in

the next E-step. The necessary expectations in these two equations are evaluated using (5.30).

In the following two sections, we apply the proposed EM–PF method to solve the *bearing-only tracking* problem with uncertain model parameters, and the so-called *sensor registration* problem in a multi-sensor scenario.

5.4 Bearing-only Tracking

5.4.1 Problem Statement

We apply the EM-PF algorithm to a bearing-only target tracking problem in the presence of sensor bias. Even though known structure which may be exploited does exist in the observation model $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{\theta})$ in this case, here we choose to ignore it, and model $\boldsymbol{h}(\cdot)$ as a mixture of Gaussians. This is done to demonstrate that useful state information can be estimated with limited knowledge of the model.

The problem consists of a linear state transition and a nonlinear measurement process. The problem is defined in [13]. In this scenario, a platform with a sensor moves according to the discrete time equations:

$$x_p(t) = \overline{x}_p(t) + \Delta x_p(t), \quad y_p(t) = \overline{y}_p(t) + \Delta y_p(t), \qquad t = 1, 2, \ldots \tag{5.31}$$

where $\overline{x}_p(t)$ and $\overline{y}_p(t)$ are the average platform position coordinates, and the perturbations $\Delta x_p(t)$ and $\Delta y_p(t)$ are assumed to be mutually independent zero-mean Gaussian white noise sequences with variances r_x and r_y, respectively. The average (unperturbed) platform motion is assumed to be horizontal with constant velocity. Its position as a function of the discrete time t (in meters) is:

$$\overline{x}_p(t) = a_1 t, \quad \overline{y}_p(t) = a_2 \tag{5.32}$$

where a_1 and a_2 are constants.

It is assumed a target moves on the $\boldsymbol{x}$-axis according to

$$\boldsymbol{x}(t+1) = \boldsymbol{F}(t)\boldsymbol{x}(t) + \boldsymbol{w}(t) \tag{5.33}$$

where:

$$\boldsymbol{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}, \quad \boldsymbol{F}(t) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \tag{5.34}$$

and x_1 and x_2 denote the position and velocity of the target, $T = 1s$ is the normalized sampling period, and $\boldsymbol{w}(t) \sim N(\boldsymbol{0}, \boldsymbol{\Sigma}_w)$, where

$$\boldsymbol{\Sigma}_{\boldsymbol{w}} = q \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix} \tag{5.35}$$

and q is a scalar. The sensor measurement process is:

$$z(t) = h(\boldsymbol{x}_p(t), \boldsymbol{y}_p(t), x_1(t)) + \beta + v_s(t), \tag{5.36}$$

where

$$h(\boldsymbol{x}_p(t), \boldsymbol{y}_p(t), x_1(t)) = \tan^{-1} \frac{y_{1p}(t)}{x_1(t) - x_{1p}(t)} \tag{5.37}$$

is the bearing between the horizontal and the line of sight from the sensor to the target, and the sensor noise $v_s(t)$ is zero mean white Gaussian with variance r_s. The sensor noise is assumed to be independent of the sensor platform perturbations. Also β is the unknown bias of the measurements.

The estimation of the target's state is performed using *only* the measurements (5.36).

The platform location perturbations induce additional errors in the measurements. The effect of these errors is evaluated by expanding the nonlinear measurement function h in a Taylor series about the average platform position. The resulting measurement process can then be written as:

$$z(t) = h(\overline{\boldsymbol{x}}_p(t), \overline{\boldsymbol{y}}_p(t), x_1(t)) + v(t) = \tan^{-1} \frac{\overline{y}_{1p}(t)}{x_1(t) - \overline{x}_{1p}(t)} + \beta + v(t), \tag{5.38}$$

where the equivalent measurement noise $v(t)$ is zero mean white Gaussian with variance given by

$$\mathrm{E}[v(t)^2] \triangleq r(t) = \frac{(\overline{y}_{1p}(t))^2 r_x + (x_1(t) - \overline{x}_{1p}(t))^2 r_y}{\{(\overline{y}_{1p}(t))^2 + (x_1(t) - \overline{x}_{1p}(t))^2\}^2} + r_s(t). \tag{5.39}$$

Notice that the variance of the equivalent measurement noise is time varying. For more details on modeling the new measurement process refer to [13].

In the following, we use the EM-PF algorithm to track the target corresponding to the uncertain observation model which has been discussed. This method ignores any known structure in the model. No doubt better performance could be achieved if a method which exploits the model structure of (5.38) were used, where $\overline{y}_{1p}(t), \overline{x}_{1p}(t)$ and β were treated as unknown parameters. Despite this fact, this example successfully demonstrates that the EM-PF method can be applied to the problem of bearing–only tracking with model uncertainty in the form of sensor bias. The example also demonstrates that the EM-PF method can be successfully applied to a range of problems where little is known about the structure of the observation model.

5.4.2 Simulation Results

In this simulation scenario, the parameter values are listed in Table 5.1. The measurements are biased by a value of $\beta = 0.5$ radians (see Figure 5.2).

It is important to compare the performance of the optimal smoother and that of the proposed EM–PF algorithm. In general, the optimal smoother is analytically intractable or prohibitively expensive to run. However, even if the optimal smoother is impossible to run, is still possible to determine the performance of the optimal smoother in terms of some performance criterion such as the mean–squared error (MSE). Indeed, if we resort to the computation of the posterior Cramér-Rao lower bound (PCRLB) as described in [16, 95], we can determine the achievable MSE of the

Table 5.1: Parameters for the bearing–only tracking simulation example.

Parameter symbol	Meaning	Value
r_x	variance of $\Delta \boldsymbol{x}_p$ in (5.31)	$1m^2$
r_y	variance of $\Delta \boldsymbol{y}_p$ in (5.31)	$1m^2$
a_1	see (5.32)	4 m/sec
a_2	see (5.32)	20 m
q	covariance scalar in (5.35)	0.01 m^2/sec^3
r_s	measurement noise variance in (5.38)	$5.24 \times 10^{-3}\text{rad}^2$
$\boldsymbol{x}_0$	initial condition for the state	$[80, 1]^T$
P	Number of Gaussian kernels in the MoG model	20
L	Number of observations	21
N	Number of particles	200
J	Number of sensors	1
M	Number of state variables	2

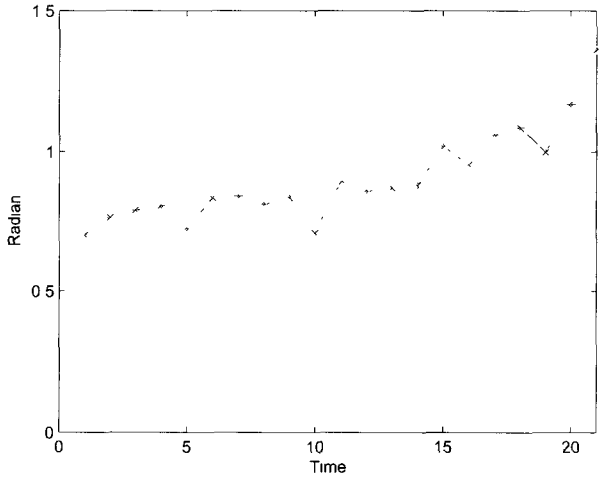


Figure 5.2: Unbiased, noise-free but perturbed measurements obtained from (5.36) with $\beta = 0$ (bottom), and the biased, perturbed, noisy measurements from (5.38), that are input to the algorithm (top).

generally intractable optimal smoother. More importantly, we can obtain a theoretical benchmark for any other practical suboptimal smoothing algorithm.

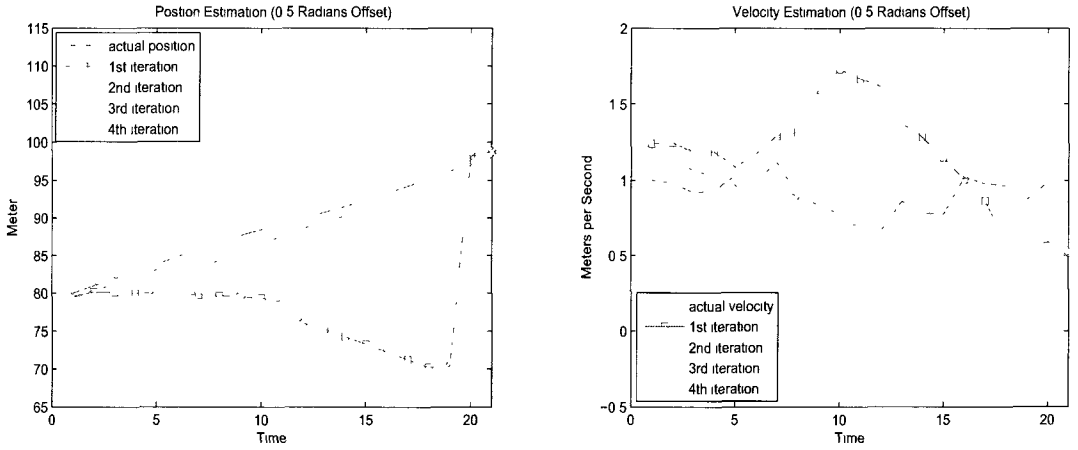
Formally, the PCRLB for *fixed–interval smoothing* can be stated as follows:

$$\boldsymbol{M}_k = \mathrm{E}\left[(xk - \hat{\boldsymbol{x}}_k(t))(xk - \hat{\boldsymbol{x}}_k(t))^T\right] \geq \boldsymbol{J}_k^{-1} \tag{5.40}$$

where $\boldsymbol{M}_k$ is the MSE correlation matrix and $\boldsymbol{J}_k^{-1}$ denotes a matrix which can be recursively computed as described in [16]. We stess that $\hat{\boldsymbol{x}}_k(t)$ need not be an unbiased estimator, and that (5.40) is a matrix inequality in the sense that $\boldsymbol{M}_k - \boldsymbol{J}_k^{-1}$ is a positive semi-definite matrix. In general, (5.40) provides a lower bound on the MSE of the considered estimator $\hat{\boldsymbol{x}}_k(t)$.

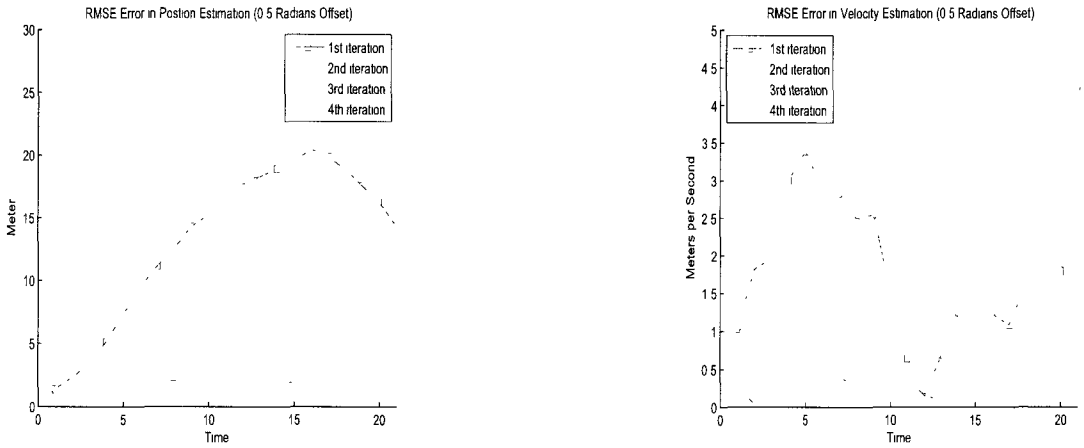
Figure 5.3 shows the position and the velocity tracking trajectories, respectively, over four successive iterations of the EM-PF algorithm for the bearing-only tracking problem for a typical run. Also, Figure 5.4 shows the root MSE error for tracking the position and velocity of the target over 50 Monte-Carlo runs, respectively. The figures represent the error for four consecutive iterations of the algorithm.

Figure 5.5 shows the position and velocity root MSE's of the EM-PF algorithm at the fourth iteration for the same run. Also shown are the corresponding PCRLB curves. In this case, the PCRLB assumes the model is known, except that the biases are unknown random variables, constant over the observation interval. It is noted that the performance of the EM-PF root MSE is worse than the PCRLB. This discrepancy is to be expected, since the PCRLB results correspond to a known observation model, (except for the bias), whereas the EM-PF algorithm assumes no knowledge of the model. Also, there are errors in the MoG and particle filter approximations. As can be seen, the EM-PF algorithm is capable of managing uncertain dynamics by identifying them, and then using this information for better estimation of the states. Although the simulation results are provided for Gaussian nonlinear measurement dynamics, the EM–PF algorithm is nevertheless capable of handling the non–Gaussian case.



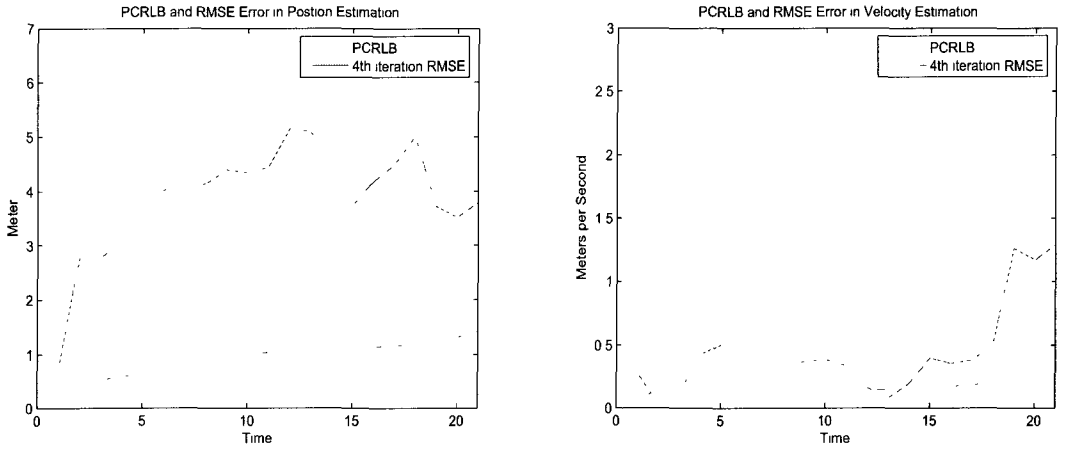
(a) (b)

Figure 5.3: Position (a) and velocity (b) tracking trajectories for the EM-PF algorithm over four successive iterations.



(a) (b)

Figure 5.4: Root MSE of the position (a) and velocity (b) state estimates vs. time over 50 Monte–Carlo runs of the EM-PF algorithm for four iterations.



(a) (b)

Figure 5.5: Root MSE of the position (a) and velocity (b) state estimates vs. time of the EM-PF algorithm at the fourth iteration, along with the corresponding PCRLB curves.

We also present results in Figure 5.6 showing root MSE vs. observation noise variance (q in (5.35)) averaged over 50 simulation runs. We can see that the root MSE increases relatively smoothly with increasing noise variance until a threshold is reached at a noise variance value of about 10^{-2}, beyond which the method breaks down. The root MSE does not steadily decrease to zero with decreasing noise variance, due to the errors in the MoG model and the error in the particle filter approximations.

We now compare the performance obtained from the EM-PF algorithm with the interactive multiple model (IMM) approach [13] for the same bearing–only tracking problem, where each model uses an extended Kalman filter (EKF) with a different range of bias[3]. In the following experiments, we used 16 models, whose corresponding bias values are uniformly distributed over 0 to 1 radian. The various parameters

[3]Experiments were also conducted using a conventional EKF, using a model which did not incorporate bias. In this case, the track loss rate approached 100%.

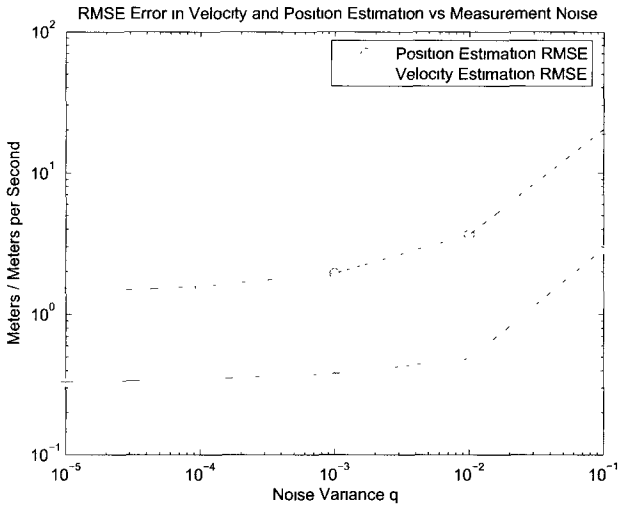


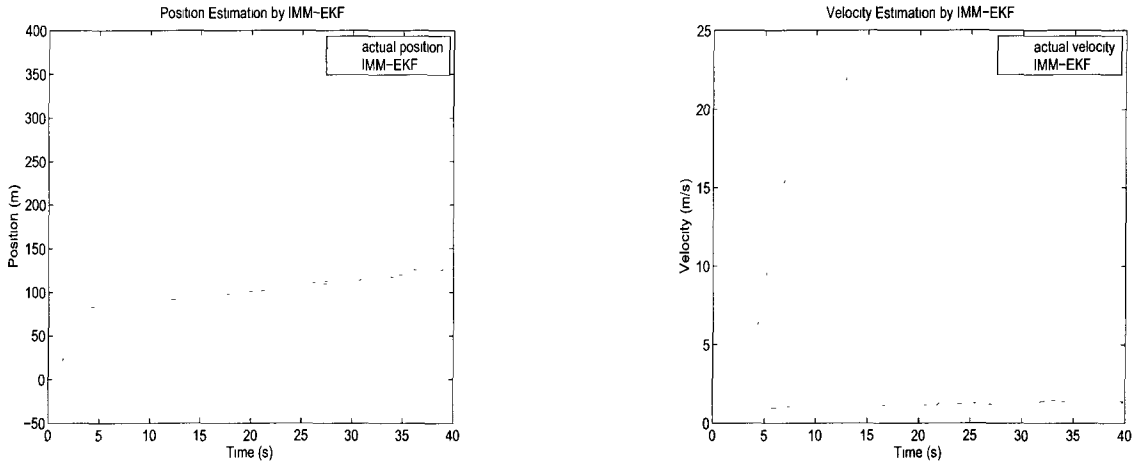
Figure 5.6: Root MSE vs. observation noise variance for the bearing–only tracking problem.

describing the problem were the same as those for the EM–PF case, and are given in Table 5.1.

Figure 5.7 shows the position and velocity estimation performance for the IMM algorithm. The corresponding RMSE's are shown in Figure 5.8 over the time interval $[0, 40]$s. The RMSE's for the interval $[20, 40]$s are shown in Figure 5.9.

It may be observed that, over the interval $[20, 40]$s after which the IMM method has acquired track, the performance of the EM–PF and IMM algorithms are roughly equivalent. However, it may be observed that the IMM model requires about 20 time steps to acquire track, because of the time required to assess the individual model probabilities and determine the winner. However, the EM-PF approach, due to its MCMC initialization procedure, requires virtually no time for acquisition. Further, any approach using EKFs must have available an accurate observation model, which is not required for the EM-PF method.

It is straightforward to modify the proposed EM–PF algorithm so that it can



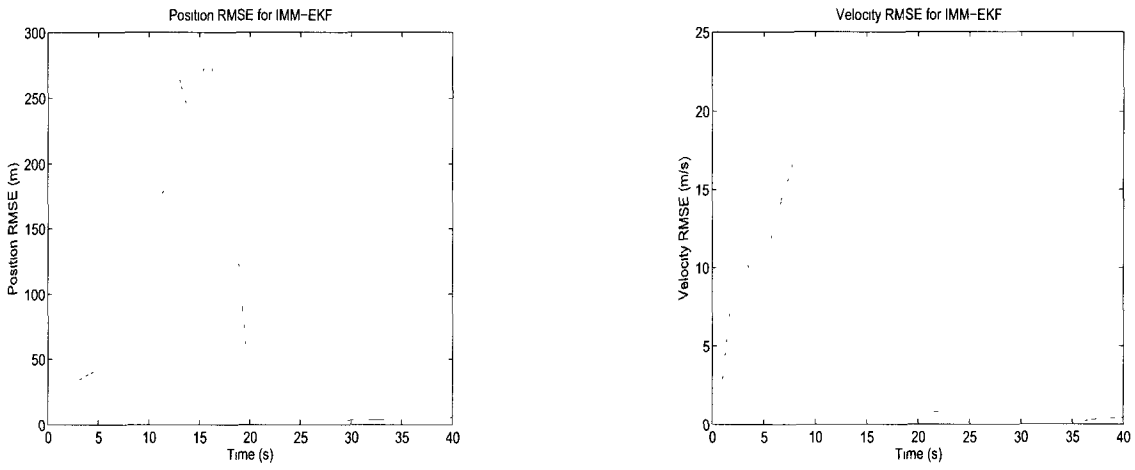
(a) (b)

Figure 5.7: Position and velocity estimates for the IMM-EKF algorithm, applied to the bearing-only tracking problem.

handle time variations in the model parameters $\boldsymbol{\theta}$. In this vein, a hypothetical experiment was conducted where the bias changed sign half–way through the observation record. It was observed that the EM-PF method quickly adapted to this change in model parameters.

5.5 Sensor Registration

We first introduce the fundamental idea of the sensor registration problem. An example of this problem is in the tracking scenario where multiple targets are being tracked by multiple sensors. The locations of the sensors are determined by a Cartesian coordinate system, while measurements from the sensors are obtained in polar coordinates. To properly combine the measurements in a multisensor scenario, it is required to transform the measurements into a common reference frame free from sensor registration errors. In a multi–sensor scenario, sensor registration errors can



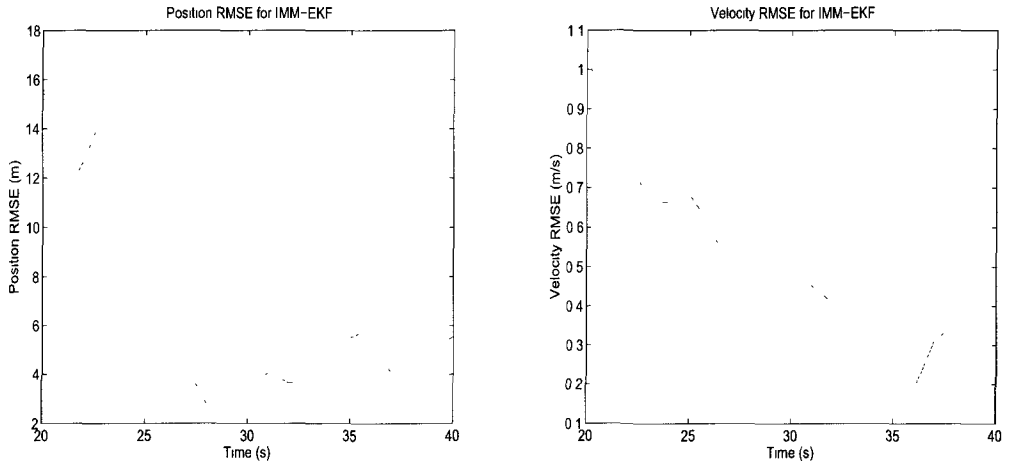
(a) (b)

Figure 5.8: Position and velocity estimate RMSEs for the IMM-EKF algorithm, applied to the bearing-only tracking problem over the time interval 0 to 40s.

cause significant error in the target location. Biased measurements, for example, can increase estimation error or even corrupt the estimation process completely.

Bias estimation is inevitable in current multisensor estimation scenarios. The classical approach to mitigate this problem is to firstly transform the measurements into a common coordinate system, estimate the biases by a batch algorithm and then remove the bias from the subsequent measurements. The EM-PF algorithm can be applied in this regard. The EM-PF algorithm may be considered similar to the recently reported method called maximum likelihood registration (MLR) [81] that indirectly estimated sensor biases and removes the effect of them in the estimation process.

In surveillance applications, it is known that the *stereographic projection* of three dimensional data onto a two-dimensional plane introduces error in sensor registration [81]. We overcome this problem using geodetic transformations for mapping the sensor measurements into the earth centered earth fixed (ECEF) coordinate system.



(a) (b)

Figure 5.9: Position and velocity estimate RMSEs for the IMM-EKF algorithm, applied to the bearing-only tracking problem, for the time interval from 20s to 40s.

Sensor registration is then performed in the ECEF coordinate system. The performance of the EM-PF algorithm is determined using simulations based on a scenario presented previously in [81].

5.5.1 Problem Statement

The problem definition provided in this section is based on the presentation in [81]. The state vector $\boldsymbol{x}(t)$ of a moving target at time t consists of the three-dimensional position of the target defined in ECEF coordinate system:

$$\boldsymbol{x}(t) = [X(t) \;\; Y(t) \;\; Z(t)]^T. \tag{5.41}$$

The origin of the ECEF coordinate system is at the center of the Earth. The X axis is extends from origin to the intersection of the prime meridian (0^o longitude) and the equator (0^o latitude). In the right-handed coordinate system, the Y axis extends from the origin to the intersection of the 90^o longitude and the equator. Also

the Z axis passes through the origin and the north pole (90^o latitude).

Consider M sensors located at $(L_m, \lambda_m, \alpha_m)$ $(m = 1, ..., M)$, where L_m is the geodetic latitude, λ_m is the longitude and α_m is the altitude above the reference ellipsoid, in the *geodetic* coordinate system. At time instant t, the mth sensor measures the position of a common target in terms of a three-dimensional measurement vector $\boldsymbol{z}_m(t)$[4]:

$$\boldsymbol{z}_m(t) = [\rho_m(t) \;\; \gamma_m(t) \;\; \epsilon_m(t)]^T, \qquad m = 1, \ldots, M, \tag{5.42}$$

where M is the number of sensors, ρ_m is slant range, γ_m is azimuth (measured clockwise from North), and ϵ_m is elevation, each with respect to the mth sensor. The registration vector for each sensor also consists of the corresponding biases, i.e.,

$$\boldsymbol{\beta}_m = [\Delta\rho_m \;\; \Delta\gamma_m \;\; \Delta\epsilon_m]^T. \tag{5.43}$$

In order to register the sensor measurements in a common coordinate system, we transform the sensor position data into the ECEF coordinate system. Given the sensor position $(L_m, \lambda_m, \alpha_m)$ the following equations give the ECEF coordinates (X_m, Y_m, Z_m):

$$X_m = (c + \alpha_m).\cos L_m.\cos\lambda_m \tag{5.44}$$

$$Y_m = (c + \alpha_m).\cos L_m.\sin\lambda_m \tag{5.45}$$

$$Z_m = (c(1 - e^2) + \alpha_m).\sin L_m, \tag{5.46}$$

in which we adopt an $WGS - 84$ ellipsoid[5] with parameters:

$$c = a/\sqrt{1 - e^2 \;\sin^2 L_m} \tag{5.47}$$

$$e = 0.0818 \tag{5.48}$$

$$a = 6378137.0 \text{ m.} \tag{5.49}$$

[4]Note that the adopted notation implies that non–time varying coordinates specify *sensor* locations, whereas time varying coordinates specify *target* locations.

[5]The world geodetic system (1984) (WGS-84) is a standard for earth coordinate systems. The WGS-84 ellipsoid minimizes the error between itself and the true shape of earth over a specific region of interest.

Since the state vector consisting of the position of the target (as well as the position of sensors) is defined in ECEF coordinates, and the measurements are in polar coordinates, it is difficult to write the explicit dependence of the measurement functions h on the state vector. Instead, we proceed to transform the state vectors into polar coordinates and model the measurement process in this system. Define the target state vector in the local tangent plane of sensor m as:

$$\boldsymbol{x}_m(t) = [\varepsilon_m(t) \;\; \nu_m(t) \;\; \upsilon_m(t)]^T, \tag{5.50}$$

where $\varepsilon_m(t)$, $\nu_m(t)$ and $\upsilon_m(t)$ denote east, north, and up axes at sensor m. These components are computed in terms of the state vector given by (5.41) and the position of the sensors as follows:

$$\begin{aligned}
\varepsilon_m(t) &= -(X(t) - X_m)\sin\lambda_m + (Y(t) - Y_m)\cos\lambda_m, \\
\nu_m(t) &= -(X(t) - X_m)\sin L_m \cos\lambda_m - (Y(t) - Y_m)\sin L_m \sin\lambda_m + (Z(t) - Z_m)\cos L_m \\
\upsilon_m(t) &= (X(t) - X_m)\cos L_m \cos\lambda_m + (Y(t) - Y_m)\cos L_m \sin\lambda_m + (Z(t) - Z_m)\sin L_m.
\end{aligned}$$

Now we can express the nonlinear measurement functions (h_{mj}; $m = 1, ..., M$; $j = 1, 2, 3$ for sensor m and the three measurement components, in terms of the state vector (5.50) as follows:

$$h_{m1}\left(\boldsymbol{x}_m(t)\right) = \sqrt{\varepsilon_m^2(t) + \nu_m^2(t) + \upsilon_m^2(t)}, \tag{5.51}$$

$$h_{m2}\left(\boldsymbol{x}_m(t)\right) = \tan^{-1}\left(\frac{\varepsilon_m(t)}{\nu_m(t)}\right), \tag{5.52}$$

$$h_{m3}\left(\boldsymbol{x}_m(t)\right) = \sin^{-1}\left\{\frac{\upsilon_m(t)}{\sqrt{\varepsilon_m^2(t) + \nu_m^2(t) + \upsilon_m^2(t)}}\right\}. \tag{5.53}$$

Having prepared the necessary definitions and assuming that the location of the static sensors are known perfectly, we can now define the measurement process for sensor m as:

$$\boldsymbol{z}_m(t) = \boldsymbol{h}_m\left(\boldsymbol{x}_m(t)\right) + \boldsymbol{\beta}_m + \boldsymbol{v}_m(t), \qquad m = 1, \ldots, M, \tag{5.54}$$

where $\boldsymbol{z}_m(t) \in \Re^{3\times 1}$ consists of the three measurement components range (ρ), azimuth (γ), and elevation (ϵ) of the target, respectively. Also, $v_m \in \Re^{3\times 1}$ is the random measurement noise vector assumed to be *i.i.d.* white noise, mutually independent from component to component, with covariance matrix $\boldsymbol{\Sigma}_{z_m} = \text{diag}(\sigma^2_{\rho_m}, \sigma^2_{\gamma_m}, \sigma^2_{\epsilon_m})$. The nonlinear functions $h_{mj},\ j = 1, 2, 3$, are defined in (5.51) – (5.53). For M sensors measuring the location of the common target, the measurements from (5.54) can be combined into a single equation as follows:

$$\boldsymbol{z}(t) = \boldsymbol{h}\left(\boldsymbol{x}(t)\right) + \boldsymbol{\beta} + \boldsymbol{v}(t), \tag{5.55}$$

where $\boldsymbol{v}(t) = [v_1(t)^T ... \ v_M(t)^T]^T$ is the random measurement noise vector assumed to be an *i.i.d.* white noise process, mutually independent from sensor to sensor. The measurement covariance matrix is $\boldsymbol{\Sigma}_z = \text{diag}(\Sigma_{z_1},\ ...,\ \Sigma_{z_M}) \in \Re^{3M\times 3M}$. The vectors $\boldsymbol{x}(t) = [\boldsymbol{x}_1(t)^T ... \ \boldsymbol{x}_M(t)^T]^T$ and $\boldsymbol{z}(t) = [\boldsymbol{z}_1(t)^T ... \ \boldsymbol{z}_M(t)^T]^T$ are the state variable and the noisy output measurement vectors for the M sensors, respectively. Also, the vector valued nonlinear function $\boldsymbol{h}_{mj}, m = 1, ..., M; j = 1, 2, 3$ is assumed to be known for the M sensors and the three values of the measurement vectors $\boldsymbol{z}_m(t) = [\rho_m(t) \ \ \gamma_m(t) \ \ \epsilon_m(t)]^T$. The vector $\boldsymbol{\beta} = [\beta_1^T ... \ \beta_M^T]^T \in \Re^{3M}$ contains the unknown biases for the M sensors that is assumed to be deterministic, time-invariant and independent of the state vector $\boldsymbol{x}(t)$.

The state process is assumed to be modelled by a linear first-order Markov process as follows:

$$\boldsymbol{x}(t+1) = \boldsymbol{x}(t) + \boldsymbol{w}(t), \tag{5.56}$$

where $\boldsymbol{w}(t) \in \Re^3$ is an *i.i.d.* noise process with covariance matrix $\boldsymbol{R} = \text{diag}(\sigma_x,\ \sigma_y,\ \sigma_z)$, and $\boldsymbol{x}(t) = [X(t) \ \ Y(t) \ \ Z(t)]^T$ is position of the target in the ECEF coordinate system.

Given L measurement vectors $\boldsymbol{z}(t),\ t = 1, \ldots, L$, the problem is to remove the effect of the measurement biases $\boldsymbol{\beta}$ and to estimate the states $\boldsymbol{x}(t),\ t = 1, \ldots, L$

accurately.

5.5.2 Simulation Results

We implement exactly the same scenario in [81] to compare the performance of the EM-PF algorithm with the recently reported MLR algorithm for sensor registration example. The details of the simulation scenario are given here from the stated reference: There are two ground-based sensors measuring the position of a moving target. The geodetic coordinate of sensors, $(L_m, \lambda_m, \alpha_m)$, are: $(-12^o30', 131^o6', 15\text{m})$ for sensor 1 and $(-14^o18', 129^o36', 10\text{m})$ for sensor 2. The target is flying from geodetic coordinate $(-12^o, 129^o30', 10\text{km})$ to $(-13^o30', 130^o30', 10\text{km})$, then it makes a mild turn and finished at $(-14^o, 131^o12', 10\text{km})$. A total of $K = 120$ synchronous pairs of measurements are collected. We assign $\sigma_x = \sigma_y = \sigma_z = 10^2$ m^2/s^4.

The true sensor biases used in simulations are as follows. Sensor 1: $\Delta\rho_1 = 2.5$ km; $\Delta\gamma_1 = -2.5^o$; $\Delta\epsilon_1 = -0.5^o$. Sensor 2: $\Delta\rho_2 = -1.8$ km; $\Delta\gamma_2 = 3^o$; $\Delta\epsilon_2 = 1^o$. Measurement noise is zero-mean Gaussian with covariance $\Sigma_{z_m} = \text{diag}(\sigma^2_{\rho_m}, \sigma^2_{\gamma_m}, \sigma^2_{\epsilon_m})$, for $m = 1, 2$. The standard deviations of the measurement noise used in the simulations are [81]: $\sigma_{\rho_1} = \sigma_{\rho_2} = 100$ m; $\sigma_{\gamma_1} = \sigma_{\gamma_2} = 0.2^o$ and $\sigma_{\epsilon_1} = \sigma_{\epsilon_2} = 0.25^o$.

Figure shows the true trajectory of the target as well as the initial estimates of the target position by the two sensors in geodetic coordinates. The differences between these trajectories are the result of unknown bias values for the sensors. Figure (a) shows these trajectories after the application of the EM-PF algorithm for sensor registration. It can be seen from this figure that the EM-PF algorithm is capable of compensating for the effect of the bias errors in track estimation after four iterations.

Figure (b) shows the RMS error of the position estimation for 100 Monte Carlo runs of the EM-PF algorithm. The algorithm is successful in compensating the effect of the unknown bias terms existing in the two sensors. It can be seen from the figure that the estimation error converges to a small value after four iterations.

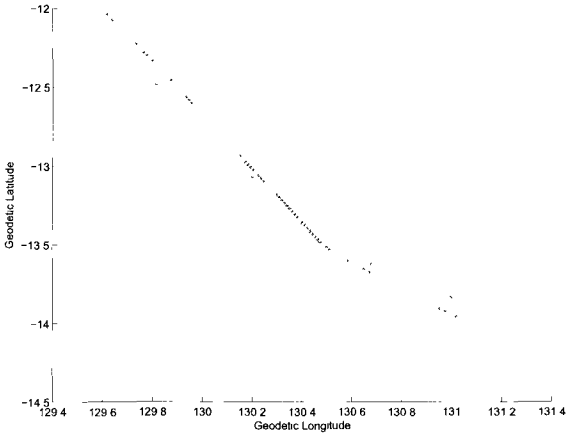
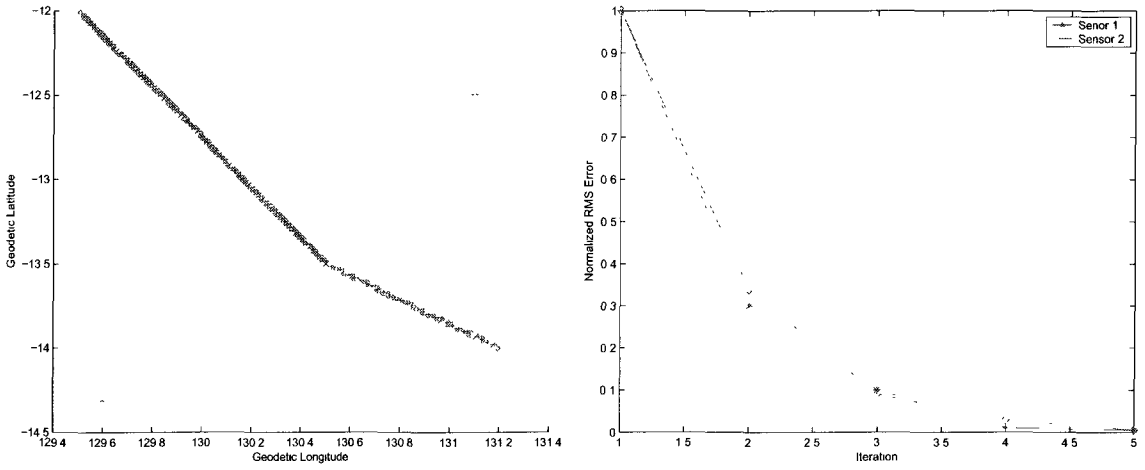


Figure 5.10: True (circle) and biased (dot) target trajectories estimated by two sensors (star)

The performance of the EM–PF algorithm is virually identical to that of the MLR method, shown in [81] for the same simulation scenario. However, unlike the MLR method, the EM–PF algorithm can be applied to non–Gaussian noise. Further, the EM–PF method is general technique, applicable to a wide range of problems, which include linear or nonlinear models and Gaussian or non–Gaussian noise in the presence of model uncertainty.

5.6 Conclusions

An EM-type algorithm for solving a joint estimation-identification problem for nonlinear non-Gaussian state-space estimation when the observation model is uncertain, is proposed. The expectation (E) step is implemented by a particular type of particle filter that is initialized by a Monte-Carlo Markov chain algorithm. Within this step, the posterior distribution of states given the measurements as well as the state vectors are estimated. Consequently, in the maximization (M) step, the nonlinear measurement process parameters are approximated using a nonlinear regression method for adjusting the parameters of a mixture of Gaussians (MofG) model. The model parameters



(a) (b)

Figure 5.11: (a)True and registered target trajectories after application of the EM-PF algorithm, and (b) RMS position error for two sensors vs. iteration number

are determined by solving a linear system of equations. The proposed method, which we refer to as the EM-PF algorithm, is used to solve a highly nonlinear bearing-only tracking problem with uncertain (biased) measurements. It is shown that the algorithm is capable of accurately tracking the state vector while identifying the unknown measurement dynamics. Also, the EM-PF algorithm is applied to solving a sensor registration problem in a multisensor fusion scenario. It is shown that the algorithm is successful in compensating the effect of unknown bias terms existing in the sensors in the target tracking scenario.

By using a nonlinear regression method based on fitting a mixture of Gaussians to the observations, the algorithm is capable of approximating a wide range of nonlinearities in the measurement and state transition processes. Also, implementing the E-step with a particle filter provides the possibility of employing the algorithm in the presence of non-Gaussian noise, e.g., with impulsive or multi-modal distributions.

Chapter 6

Conclusions

6.1 Conclusions

The main theme of this dissertation is statistical estimation. We studied three different but related applications of iterative estimation algorithms. In what follows, we briefly highlight the main contributions of the thesis in chronological order of study.

6.1.1 Joint identification and estimation using Bayesian filters and MCMC

The application of simulation-based Bayesian estimation methods (e.g. particle filters and MCMC methods) were studies to extend the role of the EM algorithm for joint estimation and identification in non-linear state estimation problems. The proposed algorithm was a variation of the *expectation-maximization* (EM) algorithm. The E-step was implemented by a particle smoother whose importance distribution was initiated by the MCMC sampling, and the maximization was a nonlinear regression using a mixture of Gaussian kernels. We solved two nonlinear problems to examine the performance of the algorithm– a *bearing-only tracking problems with uncertainty*

in measurement model and *registration in multi-sensor fusion* [108] [107]. The results compared to previously proposed methods like IMM and EKF and showed superior performance.

The resulting iterative estimation algorithms were used to tackle an unconventional joint estimation and identification non-linear filtering problem. On one hand, using a particle-filter initialized by the MCMC algorithm provided the possibility of dealing with non-Gaussian noise. On the other hand, a nonlinear regression method simplified the problem of dealing with a nonlinear measurement model. The proposed algorithm implemented a general solution for joint estimation and identification in nonlinear systems.

6.1.2 Wireless MIMO blind channel estimation

The application of iterative methods for estimation with incomplete data from a geometrical point of view was studied, with application to wireless channel estimation. Our study of the *information geometry* (differential geometric approach to statistics) resulted in a novel and extremely fast algorithm for semi-blind MIMO channel estimation [112][109]. A simple approximation of the input constellation space in a MIMO wireless communications system by a Gaussian distribution was used with this approach. This approximation, when combined with a double-projection on the set of probability distributions, led to a major improvement in the quality of channel estimation. We improved the speed of previously proposed methods by a considerable margin, while the performance of our algorithm was comparable to previous approaches [113]. This experience showed the importance of using sophisticated geometrical tools in estimation.

6.1.3 Distributed estimation

The investigation of iterative estimation algorithms for distributed estimation resulted in new contributions in information theory, relating to study of sufficient and necessary rates in distributed estimation. It also opened new insights in using low-complexity linear codes for distributed estimation.

Sufficient and necessary rates: A major part of the third problem considered was an effort to answer the following question "what if the purpose of communications in a distributed environment is parameter estimation rather than source reconstruction?" The first question that needed to be answered and perhaps the most obscured was: "whether the code rates in distributed estimation are different than the conventional communications (e.g. Slepian-Wolf rates)?" Surprisingly, the answer was found to be negative as reported recently [114]. The proofs of the theorems were based on "large deviation theory" (more specifically Sanov's Theorem) and "the method of types". The idea was to employ a *universal* coding scheme by *distributed binning*, and to compute the sufficient and necessary rates for transmitting the joint-type of the pair of sequences.

Determination of the region of achievable rates for efficient estimation of a general source is an extremely difficult problem. This fact is the motivation for proposing methods that provide practical guidelines for designing distributed estimation systems. One example of such approaches proposed in [88] for binary symmetric sources. Given any source parameter for binary symmetric source, this theorem determines the region of achievable rates for efficient estimation of the source. We generalized this theorem for a larger class of sources. More specifically, we provide a lower bound on the region of achievable rates (i.e. existence of encoder/decoders for attaining an accuracy equivalent to local estimation) for estimation of sources with a convex mutual information with respect to the unknown parameter θ [110].

LDPC-based distributed estimation with side–information With a given

set of rates, the next important issue was efficient implementation of universal coding schemes for distributed estimation. Since the correlation channel between the sources was assumed to be unknown at the joint decoder, the previously proposed distributed coding schemes were not useful for our purpose. We therefore extended the LDPC-based coset-coding schemes to the case where the correlation channel were unknown at the decoder. The basic ideas was to implement the expectation–maximization algorithm on a factor–graph that includes an LDPC decoding mechanism [115] [111].

6.2 Suggestions for further investigations

6.2.1 Estimation rate–distortion Theory:

One of the open questions in multiterminal estimation theory is how the theory relates to rate-distortion theory. The most recent works in the field provide a theoretical framework for the limits of accuracy in distributed estimation when positive rates are available. The results, however, do not provide a clear and immediate relationship with rate-distortion theory, for the following reasons. Firstl, the region of achievable rates are functions of the unknown parameters. Therefore, the region shifts when the parameter changes, and thus no immediate relation between the region or rates for efficient estimation and the region of rates in distributed coding can be established. Secondl, the region of rates as well as the attainable accuracy in estimation are functions of the test channels chosen. It has been shown in [48] that any particular selection of the test channels might result in a different region of rates as well as different limits in attainable accuracy. This is in contrast to the conventional rate-distortion theory where the region of rates for communications is unique. Therefore, up to this date, there is no obvious methodology for designing the best estimator with the best possible accuracy int he distributed estimation context.

Therefore, an interesting direction for future research is to bridge the gap between distributed estimation and rate–distortion theory. The main ideas can be related to an appropriate measure of distortion in sequences, e.g.; their marginal or joint-type, or loss in Fisher information. This derivation can provide an engineering tool for designing communication systems for the purpose of distributed estimation.

6.2.2 (Network) Information theory and statistical analysis

Recent advances in information technology have motivated extensive research on network information theory and distributed (e.g. collaborative) communications. As network technology advances, special purpose communications systems for the statistical analysis of distributed signals and data will emerge. This requires a broad range of theoretical and applied research, some of which are as follows.

Network information theory and coding: It is known that the Shannon separation theorem does not hold in general for networks. A theoretical study of network source-channel transmission problems with a goal of optimizing distributed *universal codes* in *multiple-access* and *broadcast* networks is needed. The use of *correlated channel codes* seems to be promising for such purposes, like multi-casting over networks and collaborative communications.

On the other hand, special purpose communication systems for statistical analysis demands a different approach in design. Works on structured random codes, special purpose sparse codes (e.g. LDPC) and diversity codes (e.g. Fountain codes) for statistical analysis in networks are promising directions.

Multiterminal estimation and rate-distortion theory: Multiterminal estimation of continuous-valued sources and rate-distortion theory relating to parameter estimation are still open problems. The subject of this dissertation can be extended

for Gaussian sources, to learn more about the relationship between distributed estimation and distributed coding (e.g. Wyner-Ziv theorem). One approach is to establish estimation equations in the space of probability distributions, and using the *method of types* measure the amount of Fisher information (FI) loss under a limited entropy rate. The effect of the encoding process on sufficient statistics depends on the form of the underlying distributions and on the local behavior of the FI, notions that can be studied by information geometry and statistics.

Methods of information geometry: The applications of information geometry (IG) for statistical signal and data processing have attracted a great attention recently. Problems involve optimization over the manifolds of probability distributions, and require careful manipulation of the Fisher and/or Shannon information. For instance, consider *source-channel (network) coding*and *capacity approaching pre-coding* in MIMO systems, or *optimal sensory sampling and coding* problems. The locality of the FI (vs. the global Shannon entropy) can be used to design algorithms that behave differently in various locations of measure space, in order to optimally match to the local behavior of the source and channel. Information geometrical methods can be used to combine ideas from the *theory of loss*, *ancillary statistics* and *rate-distortion theory* to design distributed signal processing algorithms. These algorithms behave optimally in terms of the preservation of the FI, while meeting information theoretic constraints (e.g. entropy, rate, power, delay, randomness and privacy).

Appendix A

Proof of Theorem (1.1.5)

For proving the theorem we begin with some preliminaries. A statistic is defined by a mapping $h : \mathcal{X} \to \mathcal{S}$ from random variable X to a random variable $S = h(X)$. Given PD $Q(X;\theta)$, this mapping determines the PD $q(S;\theta) = q(h(X);\theta)$ governing random variable S. We define the following:

$$\begin{aligned} r(X;\theta) &= \frac{Q(X;\theta)}{q(h(X);\theta)}, \\ p(X|s;\theta) &= r(X;\theta)\delta_{h(X)}(s) \quad \text{for any fixed } s \in \mathcal{S}, \\ \Pr(A|s;\theta) &= \sum_{x \in A} p(X|Y;\theta), \quad A \subset \mathcal{X}, \end{aligned} \tag{A.1}$$

where $\delta_{h(X)}$ is the Kronecker delta function on $(\mathcal{S}, ds)$ concentrated on the point $h(X)$, and A is defined any open subset of the domain of X. Here $\Pr(A|s;\theta)$ is in fact the conditional probability of the event $\{X \in A\}$ given any given $S = s$ value of the statistic.

Using these definition, for any open subset $B \in \mathcal{S}$ we have:

$$\begin{aligned}\int_B \Pr(A|s;\theta)q(s;\theta)ds &= \int_B \sum_{x\in A} p(x|s;\theta)q(s;\theta)ds \\ &= \Pr(A \cap h^{-1}(B)) \\ &= \sum_{x\in(A\cap h^{-1}(B))} p(x;\theta),\end{aligned}$$

Therefore, if $A = \mathcal{X}$ (therefore $\Pr(A|s;\theta) = 1$ for all $s \in \mathcal{S}$), for any set $B \in S$ we have:

$$\int_B q(y;\theta) = \sum_{x\in h^{-1}(B)} Q(X;\theta).$$

For the same set $B \subset \mathcal{S}$, we use this equation to compute the following derivative with respect to θ_i:

$$\begin{aligned}E_\theta\big[\partial_i \log q(s;\theta)\big] &= \int_B \big[\partial_i \log q(s;\theta)\big]q(s;\theta)ds \\ &= \int_B \partial_i q(s;\theta)ds \\ &= \partial_i \int_B q(s;\theta)ds \\ &= \partial_i \sum_{x\in h^{-1}(B)} Q(X;\theta) \\ &= \sum_{x\in h^{-1}(B)} \partial_i l(X;\theta)Q(X;\theta) \\ &= E_\theta\big[\partial_i l(X;\theta)|h(X)\big],\end{aligned}$$

which shows that for $B \subset \mathcal{S}$:

$$E_\theta\big[\partial_i \log q(h(X);\theta)\big] = E_\theta\big[\partial_i l(X;\theta)|h(X)\big], \tag{A.2}$$

Also, for any fixed particular value of statistic $s \in B \subset \mathcal{S}$:

$$\partial_i \log q(s;\theta)\big] = E_\theta\big[\partial_i l(X;\theta)|s\big]. \tag{A.3}$$

From the assumed factorization $Q(X;\theta) = q(h(X);\theta)r(X;\theta)$ we have:

$$\partial_i l(X;\theta) = \partial_i \log q(h(X);\theta) + \partial_i r(X;\theta). \tag{A.4}$$

Therefore,

$$\begin{aligned} E_\theta\big[\partial_i r(X;\theta)|h(X)\big] &= E_\theta\big[\partial_i l(X;\theta)|h(X)\big] - E_\theta\big[\partial_i \log q(h(X);\theta)|h(X)\big] \\ &= E_\theta\big[\partial_i l(X;\theta)|h(X)\big] - E_\theta\big[\partial_i \log q(h(X);\theta)\big] \\ &= E_\theta\big[\partial_i \log q(h(X);\theta)\big] - E_\theta\big[\partial_i \log q(h(X);\theta)|h(X)\big] \quad \text{(A.5)} \\ &= 0, \quad \text{(A.6)} \end{aligned}$$

where we used the Eq. (A.2) in (A.5). This shows that $\partial_i \log r(X;\theta)$ as a function of X is orthogonal to any function of h(X) (and in particular to $\partial_j \log q(h(X);\theta)$ with respect to the *expectation inner product* defined as:

$$\langle \Phi, \Psi \rangle_\theta = E_\theta\big[\Phi(X)\Psi(X)\big].$$

We use this property when we compute the conditional covariance of the score functions given any particular *fixed* value of the statistic S as follows (we use the notation $l_\theta = \log Q(X;\theta)$ and note that $S = h(X)$):

$$\begin{aligned} Cov\big[\partial_i l_\theta, \partial_j l_\theta|s\big] &= E_\theta\Big[\big\{\partial_i l_\theta - E_\theta\big[\partial_i l_\theta|s\big]\big\}\big\{\partial_j l_\theta - E_\theta\big[\partial_j l_\theta|s\big]\big\}\,\Big|s\Big] \\ &= E_\theta\big[\partial_i l_\theta \partial_j l_\theta\big|s\big] - E_\theta\big[\partial_i l_\theta|s\big]E_\theta\big[\partial_j l_\theta|s\big] \\ &= E_\theta\big[\partial_i l_\theta \partial_j l_\theta\big|s\big] - \partial_i \log q(s;\theta)\partial_j \log q(s;\theta), \quad \text{(A.7)} \end{aligned}$$

where we used Eq. (A.3) in Eq. (A.7). We now sum up the above covariance over all

values of statistic S, we have:

$$\begin{aligned}
E_\theta\Big[Cov\big[\partial_i l_\theta, \partial_i l_\theta | S\big]\Big] &= \int \Big[Cov\big[\partial_i l_\theta, \partial_i l_\theta | s\big]\Big] q(s;\theta) ds \\
&= \int \Big[E_\theta\big[\partial_i l_\theta \partial_i l_\theta \big| s\big] - \partial_i \log q(s;\theta) \partial_j \log q(s;\theta)\Big] q(s;\theta) ds \\
&= \int \Big[E_\theta\big[\partial_i l_\theta \partial_i l_\theta \big| s\big]\Big] q(s;\theta) ds \\
&- \int \Big[\partial_i \log q(s;\theta) \partial_j \log q(s;\theta)\Big] q(s;\theta) ds \\
&= E_\theta\big[\partial_i l_\theta \partial_i l_\theta\big] - E^h_\theta\big[\partial_i \log q(s;\theta) \partial_j \log q(s;\theta)\big] \\
&= J(\theta) - J_h(\theta) \\
&= \Delta J(\theta),
\end{aligned}$$

where E^h_θ denotes the expectation with respect to the induced PD $q(s;\theta)$. We used the definition of the Fisher information of the induced distribution in the last equation. Since the covariance is always positive semidefinite $\Delta J(\theta) \geq 0$, and therefore $J_h(\theta) \leq J(\theta)$.

We now show that the equality holds when h is a sufficient statistic. We substitute from Eq. (A.4) in Eq. (A.7) we have (the PD's $q(s;\theta)$ and $r(s;\theta)$ are denoted as q_θ and r_θ, respectively):

$$\begin{aligned}
Cov\big[\partial_i l_\theta, \partial_j l_\theta | s\big] &= E_\theta\big[\partial_i l_\theta \partial_i l_\theta \big| s\big] - \partial_i \log q \partial_j \log q_\theta \\
&= E_\theta\big[\, \{\partial_i \log q_\theta + \partial_i \log r_\theta\} \{\partial_j \log q_\theta + \partial_j \log r_\theta\} \,\big| s\big] \\
&- \partial_i \log q_\theta \partial_j \log q_\theta
\end{aligned}$$

that results into:

$$\begin{aligned} Cov\big[\partial_i l_\theta, \partial_j l_\theta | s\big] &= E_\theta\big[\partial_i \log q_\theta \partial_j \log q_\theta \big| s\big] + E_\theta\big[\partial_i \log r_\theta \partial_j \log r_\theta \big| s\big] \\ &- E_\theta\big[\partial_i \log q_\theta \partial_j \log r_\theta \big| s\big] - E_\theta\big[\partial_j \log q_\theta \partial_i \log r_\theta \big| s\big] \\ &- \partial_i \log q_\theta \partial_j \log q_\theta \\ &= \partial_i \log q_\theta \partial_j \log q_\theta + E_\theta\big[\partial_i \log r_\theta \partial_j \log r_\theta \big| s\big] \\ &- 0 - 0 - \partial_i \log q_\theta \partial_j \log q_\theta \\ &= E_\theta\big[\partial_i \log r_\theta \partial_j \log r_\theta \big| s\big], \end{aligned}$$

where we used the orthogonality property proved in Eq. (A.6). This shows that the covariance vanishes when $\partial_i \log r(X;\theta) = 0$ for all θ, i, and X. In other words $\Delta J(\theta) = 0$ and the equality in achieved. This conditions is equivalent to the sufficiency of statistic $S = h(X)$.

Appendix B

Proof of Lemma (2.3.8, part (a))

Let (x_i, y_i) be $i.i.d \sim Q_0(X,Y) = Q(X)Q(Y)$. We define the joint typicality of the pair of sequences (x^n, y^n) with respect to $Q(X,Y)$ iff the sample entropies are close to their true values, i.e. for any $\epsilon > 0$:

$$\left| -\frac{1}{n} \log Q(x^n) - H(X) \right| \leq \epsilon, \tag{B.8}$$

$$\left| -\frac{1}{n} \log Q(y^n) - H(Y) \right| \leq \epsilon, \tag{B.9}$$

$$\left| -\frac{1}{n} \log Q(x^n, y^n) - H(X,Y) \right| \leq \epsilon. \tag{B.10}$$

We wish to calculate the probability (under the product distribution) of seeing a pair (x^n, y^n) that looks jointly typical of Q, i.e. (x^n, y^n) satisfies Eqs. (B.8)-(B.10). Thus (x^n, y^n) are jointly typical with respect to $Q(X,Y)$ if its joint-type is a member of the set $\mathcal{E}$, i.e. $\tilde{P}_{XY}(x^n, y^n) \in \mathcal{E} \cap \mathcal{P}_n(X,Y)$ defined as follows:

$$\begin{aligned} \mathcal{E} = \{P(X,Y) : & \left| -\sum_{x,y} P(X,Y) \log Q(X) - H(X) \right| \leq \epsilon, \\ & \left| -\sum_{x,y} P(X,Y) \log Q(Y) - H(Y) \right| \leq \epsilon, \\ & \left| -\sum_{x,y} P(X,Y) \log Q(X,Y) - H(X,Y) \right| \leq \epsilon \}. \end{aligned}$$

Using Sanov's theorem, the probability of set $\mathcal{E}$ is:

$$Q_0^n(\mathcal{E}) \leq (n+1)^{|\mathcal{X}|\times|\mathcal{Y}|} 2^{-nD(P^*\|Q_0)},$$

where P^* is the distribution closets to Q_0 in relative entropy. We wish to find $P^* \in E$ that is closest to Q_0 in KL distance. For this, we need to solve the following constraint optimization:

$$P^* = \arg\min_{P\in E} D(P \parallel Q_0),$$

For sufficiently large n we assume $\epsilon = 0$. Using Lagrange multipliers, we construct the functional:

$$\begin{aligned} J(P) &= \sum_x P(X)\log\frac{P(X)}{Q(X)} \\ &+ \lambda_0\big(\sum_{x,y} P(X,Y)\log Q(X) - H(X)\big) \\ &+ \lambda_1\big(\sum_{x,y} P(X,Y)\log Q(Y) - H(Y)\big) \\ &+ \lambda_2\big(\sum_{x,y} P(X,Y)\log Q(X,Y) - H(X,Y)\big) \\ &+ \lambda_3\sum_x P(X). \end{aligned}$$

By taking the derivative with respect to $P(X)$ and renaming the Lagrangian multipliers, the solution is in the form of:

$$P^*(X,Y) = Q_0(X,Y)\exp(\lambda_0 + \lambda_1 Q(X) + \lambda_2 Q(Y) + \lambda_3 Q(X,Y)),$$

By checking the Karush-Kuhn-Tucker (KKT) conditions ([17], page 243), it is easy to verify that this solution is unique. Moreover, by substituting $Q_0(X,Y) = Q(X)Q(Y)$ in the solution it is easy to verify that $P^*(X,Y) = Q(X,Y)$. This shows that that as $\epsilon \to 0$, P^* is the joint distribution Q and Q_0 is the product distribution.

In other words, the distribution in $\mathcal{E}$ closest to the product distribution $Q_0(X,Y) = Q(X)Q(Y)$ is the joint distribution $Q(X,Y)$. Thus:

$$\begin{aligned} \Pr(\mathcal{E}) &= Q_0^n(\mathcal{E}) \leq 2^{-nD(Q(X,Y)\|Q_0)} \\ &= 2^{-nD(Q(X,Y)\|Q(X)Q(Y))} \\ &= 2^{-nI(X,Y)}. \end{aligned}$$

Appendix C

Proof of Theorem 3.6.2

First, let:

$$\begin{aligned} Pr(x_1 = 0|y_1) = \frac{1+p}{2} \quad &\rightarrow \quad p = 2Pr(x_1 = 0|y_1) - 1 \\ Pr(x_2 = 0|y_2) = \frac{1+q}{2} \quad &\rightarrow \quad q = 2Pr(x_2 = 0|y_2) - 1 \end{aligned}$$

Then:

$$\begin{aligned} Pr(x_1 + x_2 = 0|y_1, y_2) &= Pr(x_1 = 0|y_1, y_2)Pr(x_2 = 0|y_1, y_2) \\ &+ Pr(x_1 = 1|y_1, y_2)Pr(x_2 = 1|y_1, y_2) \\ &= Pr(x_1 = 0|y_1)Pr(x_2 = 0|y_2) \\ &+ Pr(x_1 = 1|y_1)Pr(x_2 = 1|y_2) \\ &= \frac{1+p}{2}\frac{1+q}{2} + \frac{1-p}{2}\frac{1-q}{2} \\ &= \frac{1+pq}{2}, \end{aligned}$$

hence:

$$2Pr(x_1 + x_2 = 0|y_1, y_2) - 1 = pq.$$

Similarly by induction:

$$2Pr(x_1 + ... + x_n = 0|y_1, ..., y_n) - 1 = \prod_{i=1}^{n}(2Pr(x_i = 0|y_i) - 1). \tag{C.11}$$

Now since:

$$L(x_i|y_i) = \frac{Pr(x_i = 0|y_i)}{Pr(x_i = 1|y_i)} \rightarrow Pr(x_i = 0|y_i) = \frac{L(x_i|y_i)}{1 + L(x_i|y_i)},$$

we have:

$$2Pr(x_i = 0|y_i) - 1 = \frac{L_i - 1}{L_i + 1} = \tanh \frac{l_i}{2}, \tag{C.12}$$

with $L_i = L(x_i|y_i)$ and $l_i = \log L_i$. By substituting Equation (C.12) in Equation (C.11) the proof is for the case when $s_i = 0$ is complete. The proof for the case $s_i = 1$ is obvious by noticing that $Pr(x_i = 1|y_i) = Pr(x_i = 0|y_i)$ which in turn inverts the sign of the likelihoods.

Appendix D

IGID and the Variational EM algorithm

In this Appendix we prove the equivalence of the IGID and EMS algorithms by introducing a variational version of the EM algorithm originally developed in [82]. Suppose that L input-output pair of observations $(x_1, y_1), ..., (x_L, y_L)$ are available. The standard maximum likelihood estimation aims to maximize the log-likelihood of the *complete-date* defined as:

$$\mathcal{L} = \log \prod_{k=1}^{L} q(y_k, x_k) = \sum_{k=1}^{L} \log q(y_k, x_k). \tag{D.13}$$

However, since the corresponding inputs of the observations are not available in the blind ML estimation problem, the procedure is based on only the output observations,

which are also referred to as the incomplete-data:

$$\begin{aligned}
\mathcal{L} &= \log \prod_{k=1}^{L} q(y_k) \\
&= \sum_{k=1}^{L} \log q(y_k) \\
&= \sum_{k=1}^{L} \log \sum_{x} q(y_k, x) \qquad \text{(D.14)} \\
&= \sum_{y} \tilde{p}(y) \log \sum_{x} q(y, x), \qquad \text{(D.15)}
\end{aligned}$$

where the summation over x in (D.14) represents the set of all possible input values corresponding to the output observation. Also,(D.15) is obtained using the definition of the *empirical distribution* as defined in (4.5).

Using an arbitrary *variational* distribution over the input space, $u(x|y)$ we obtain a lower bound on the log-likelihood as follows:

$$\begin{aligned}
\mathcal{L} &= \sum_{y} \tilde{p}(y) \log \sum_{x} u(x|y) \frac{q(y,x)}{u(x|y)} \qquad \text{(D.16)} \\
&\geq \sum_{y} \sum_{x} \tilde{p}(y) u(x|y) \log \frac{q(y,x)}{u(x|y)} \qquad \text{(D.17)} \\
&= \sum_{y} \sum_{x} \tilde{p}(y) u(x|y) \log \frac{\tilde{p}(y) q(y,x)}{\tilde{p}(y) u(x|y)} \\
&= \sum_{y} \sum_{x} \tilde{p}(y) u(x|y) \log \frac{q(y,x)}{u(x|y)\tilde{p}(y)} + \sum_{y} \sum_{x} \tilde{p}(y) u(x|y) \log \tilde{p}(y) \\
&= \sum_{y} \sum_{x} \tilde{p}(y) u(x|y) \log \frac{q(y,x)}{u(x|y)\tilde{p}(y)} + \sum_{y} \tilde{p}(y) \log \tilde{p}(y) \\
&= \sum_{y} \sum_{x} p(y,x) \log \frac{q(y,x)}{p(y,x)} + \sum_{y} \tilde{p}(y) \log \tilde{p}(y) \\
&= -D(p \parallel q) - H(\tilde{p}(y)) \\
&\triangleq \mathcal{F}(p,q), \qquad \text{(D.18)}
\end{aligned}$$

where $H(\tilde{p}(y))$ is the entropy of the observed empirical distribution. Eq. (D.16) is

obtained from (D.17) using the *Jensen's Inequality* and the fact that the log function is convex.

The EM algorithm consists of two consecutive iterations [82] for maximizing the lower-bound $\mathcal{F}$ (D.18) as follows:

D.1 The E-Step vs. the First Projection

In the E-step, the best variational distribution over the input, $u(x|y)$ is computed to maximize the lower-bound (D.18):

$$\begin{aligned} u(x|y) &= \arg\max_{u(x|y)} \mathcal{F}(p,q) \\ &= \arg\max_{u(x|y)} -D(p \parallel q) - H(\tilde{p}(y)) \\ &= \arg\min_{p} D(p \parallel q). \end{aligned} \tag{D.19}$$

Eq. (D.19) follows from the fact that the entropy of the output empirical distribution $H(\tilde{p}(y))$ is constant.

Eq. (D.19) shows the similarity of the E-step to the *first projection* in the IGID algorithm. It is useful to observe that the solution of the first projection, i.e. $p^*(y,x) = q(x|y)\tilde{p}(y)$ achieves the equality in (D.18):

$$\begin{aligned} \mathcal{F}(p^*,q) &= -D(p^* \parallel q(y,x)) - H(\tilde{p}(y)) \\ &= -D(q(x|y)\tilde{p}(y) \parallel q(y,x)) - H(\tilde{p}(y)) \\ &= -\sum_y \sum_x q(x|y)\tilde{p}(y) \log \frac{q(x|y)\tilde{p}(y)}{q(x|y)q(y)} + \sum_y \tilde{p}(y) \log \tilde{p}(y) \\ &= -\sum_y \tilde{p}(y) \log \frac{\tilde{p}(y)}{q(y)} + \sum_y \tilde{p}(y) \log \tilde{p}(y) \\ &= \sum_y \tilde{p}(y) \log q(y) = \mathcal{L}. \end{aligned}$$

D.2 The M-Step vs. the Second Projection

The M-step of the EM algorithm maximizes the lower-bound (D.18) given the obtained distribution $u(x|y)$ in the E-step [82]. The maximization is performed with respect to the parameters of the likelihood function, which is equivalent to finding the best likelihood distribution q within the family of likelihood distribution $\mathcal{Q}$:

$$\begin{aligned} q &= \arg\max_{q} \mathcal{F}(p, q) \\ &= \arg\max_{q}(-D(p \parallel q) - H(\tilde{p}(y))) \\ &= \arg\min_{q} D(p \parallel q). \end{aligned}$$

This minimization corresponds to the *second projection* of the IGID algorithm.

Appendix E

EM-based Blind Identification Algorithms

EM methods have been developed for various blind identification problems [5, 40, 105, 54, 26, 71, 100, 68]. We summarize these the results as follows:

$$\boldsymbol{H}_{t+1} = \sum_{k=1}^{L} \overline{\boldsymbol{y}_k \boldsymbol{x}^T} (\sum_{k=1}^{L} \overline{\boldsymbol{x}\boldsymbol{x}^T})^{-1}.$$

$$\boldsymbol{\Psi}_{t+1} = \frac{1}{L} \sum_{k=1}^{L} \overline{(\boldsymbol{y}_k - \boldsymbol{H}_{t+1}\boldsymbol{x})(\boldsymbol{y}_k - \boldsymbol{H}_{t+1}\boldsymbol{x})^T},$$

where $\boldsymbol{H}_{t+1}$ and $\boldsymbol{\Psi}_{t+1}$ are the new updated estimates of the channel gain matrix and noise covariance at iteration t and where:

$$\overline{\boldsymbol{x}\boldsymbol{x}^T} = \sum_{\boldsymbol{x}\in\Omega} \boldsymbol{x}\boldsymbol{x}^T p_x(\boldsymbol{x}|\boldsymbol{y}_k; \boldsymbol{H}_t, \boldsymbol{\Psi}_t) = \frac{\sum_{\boldsymbol{x}\in\Omega} \boldsymbol{x}\boldsymbol{x}^T p_y(\boldsymbol{y}_k|\boldsymbol{x}; \boldsymbol{H}_t, \boldsymbol{\Psi}_t)}{\sum_{\boldsymbol{x}\in\Omega} p_y(\boldsymbol{y}_k|\boldsymbol{x}; \boldsymbol{H}_t, \boldsymbol{\Psi}_t)},$$

$$\overline{\boldsymbol{x}} = \frac{\sum_{\boldsymbol{x}\in\Omega} \boldsymbol{x} p_y(\boldsymbol{y}_k|\boldsymbol{x}; \boldsymbol{H}_t, \boldsymbol{\Psi}_t)}{\sum_{\boldsymbol{x}\in\Omega} p_y(\boldsymbol{y}_k|\boldsymbol{x}; \boldsymbol{H}_t, \boldsymbol{\Psi}_t)},$$

and similarly

$$\overline{(\boldsymbol{y}_k - \boldsymbol{H}_{t+1}\boldsymbol{x})(\boldsymbol{y}_k - \boldsymbol{H}_{t+1}\boldsymbol{x})^T} = \frac{\sum_{\boldsymbol{x}\in\Omega} (\boldsymbol{y}_k - \boldsymbol{H}_{t+1}\boldsymbol{x})(\boldsymbol{y}_k - \boldsymbol{H}_{t+1}\boldsymbol{x})^T p_y(\boldsymbol{y}_k|\boldsymbol{x}; \boldsymbol{H}_t, \boldsymbol{\Psi}_t)}{\sum_{\boldsymbol{x}\in\Omega} p_y(\boldsymbol{y}_k|\boldsymbol{x}; \boldsymbol{H}_t, \boldsymbol{\Psi}_t)},$$

where $\boldsymbol{H}_t$ and $\boldsymbol{\Psi}_t$ are the channel matrix and the noise covariance from the previous iteration, respectively. Also $p_y(\boldsymbol{y}_k|\boldsymbol{x}; \boldsymbol{H}_t, \boldsymbol{\Psi}_t)$ is the likelihood function obtained by using the current values of the parameters $\boldsymbol{H}_t$ and $\boldsymbol{\Psi}_t$. These results are given assuming that the input consists of a finite set of points from constellation set Ω. Also since a uniform signalling scheme for the input is considered, the prior input distribution is $p_x(\boldsymbol{x}) = \frac{1}{C^M}$ where M is the length of input vector.

Appendix F

Proof of (4.23)

Defining the *Schur complement* of $\boldsymbol{M} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}$ as $\boldsymbol{W} = Schur(\boldsymbol{M}) = \boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}$, one can obtain the inverse of $\boldsymbol{M}$ by:

$$\boldsymbol{M}^{-1} = \begin{bmatrix} \boldsymbol{W}^{-1} & -\boldsymbol{W}^{-1}\boldsymbol{B}\boldsymbol{D}^{-1} \\ -\boldsymbol{D}^{-1}\boldsymbol{C}\boldsymbol{W}^{-1} & \boldsymbol{D}^{-1} + \boldsymbol{D}^{-1}\boldsymbol{C}\boldsymbol{W}^{-1}\boldsymbol{B}\boldsymbol{D}^{-1} \end{bmatrix}.$$

Using the above, it is straightforward to derive the inverse of the block matrix $\boldsymbol{Q}$ in (4.23).

Appendix G

Proof of (4.30)

It is easy to show that for Gaussian distributions $p = \mathcal{N}(0, \boldsymbol{P} \in \Re^{d\times d})$ and $q = \mathcal{N}(0, \boldsymbol{Q} \in \Re^{d\times d})$:

$$D(p \parallel q) = \text{trace}(\boldsymbol{Q}^{-1}\boldsymbol{P}) + \log\det\boldsymbol{Q} - \log\det\boldsymbol{P} - d. \tag{G.20}$$

Substituting $\boldsymbol{Q}^{-1}$ and $\boldsymbol{P}$ from Equations (4.23) and (4.29), respectively, in (G.20) we have:

$$\begin{aligned} \text{trace}(\boldsymbol{Q}^{-1}\boldsymbol{P}) &= \text{trace}(\boldsymbol{\Psi}^{-1}\boldsymbol{P}_{11}) \\ &- 2\text{trace}(\boldsymbol{\Psi}^{-1}\boldsymbol{H}\boldsymbol{P}_{12}^T) \\ &+ \text{trace}(\boldsymbol{\Phi}^{-1} + \boldsymbol{H}^T\boldsymbol{\Psi}^{-1}\boldsymbol{H})\boldsymbol{P}_{22}. \end{aligned} \tag{G.21}$$

In addition since

$$\det\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix} = \det\boldsymbol{A}.\det(\boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B}), \tag{G.22}$$

we have

$$\det\boldsymbol{Q}^{-1} = \det\boldsymbol{\Psi}^{-1}.\det\boldsymbol{\Phi}^{-1}. \tag{G.23}$$

Therefore:

$$\log\det \boldsymbol{Q} = -\log\det \boldsymbol{Q}^{-1} = -(\log\det \boldsymbol{\Psi}^{-1} + \log\det \boldsymbol{\Phi}^{-1}). \tag{G.24}$$

Substituting (G.20), (G.21) and (G.24) in (4.28) gives (4.30).

Appendix H

Proof of (4.31) and (4.32)

From matrix algebra we have [77]:

$$\frac{\partial(\text{trace}(\boldsymbol{\Psi}^{-1}\boldsymbol{H}\boldsymbol{P}_{12}^{T}))}{\partial \boldsymbol{H}} = \boldsymbol{\Psi}^{-1}\boldsymbol{P}_{12}$$

$$\frac{\partial(\text{trace}(\boldsymbol{H}^{T}\boldsymbol{\Psi}^{-1}\boldsymbol{H}\boldsymbol{P}_{22}))}{\partial \boldsymbol{H}} = 2\boldsymbol{\Psi}^{-1}\boldsymbol{H}\boldsymbol{P}_{22}$$

$$\frac{\partial(\text{trace}(\boldsymbol{\Psi}^{-1}\boldsymbol{P}_{11}))}{\partial \boldsymbol{\Psi}^{-1}} = \boldsymbol{P}_{11}^{T}$$

$$\frac{\partial(\text{trace}(\boldsymbol{\Psi}^{-1}\boldsymbol{H}\boldsymbol{P}_{12}^{T}))}{\partial \boldsymbol{\Psi}^{-1}} = \boldsymbol{P}_{12}\boldsymbol{H}^{T}$$

$$\frac{\partial(\text{trace}(\boldsymbol{H}^{T}\boldsymbol{\Psi}^{-1}\boldsymbol{H}\boldsymbol{P}_{22}))}{\partial \boldsymbol{\Psi}^{-1}} = \boldsymbol{H}\boldsymbol{P}_{22}^{T}\boldsymbol{H}^{T}$$

$$\frac{\partial \log\det \boldsymbol{\Psi}^{-1}}{\partial \boldsymbol{\Psi}^{-1}} = \boldsymbol{\Psi}.$$

Using these equations to compute the partial derivatives necessary in the minimization of (4.30) results in (4.31) and (4.32).

Appendix I

Proof of (4.36)

By definition, $\boldsymbol{P}$ is the covariance matrix of the complete data $\boldsymbol{z} = [\boldsymbol{y}^T \boldsymbol{x}^T]^T$. Therefore the true values of the blocks $\boldsymbol{P}_{22}$ and $\boldsymbol{P}_{11}$ in (4.29) are equal to $\boldsymbol{\Phi}$, the (true) known covariance matrix of the source $\boldsymbol{x}$, and $\boldsymbol{S}$, the sample covariance matrix of the observed data $\boldsymbol{y}$, respectively. From (4.31), we have

$$\boldsymbol{P}_{12} = \boldsymbol{H}^\star \boldsymbol{\Phi}.$$

Using this result in (4.32), we have

$$\begin{aligned} \boldsymbol{P}_{11} = \boldsymbol{S} &= \boldsymbol{\Psi}^\star + \boldsymbol{H}^\star \boldsymbol{\Phi} \boldsymbol{\Phi}^{-1} \boldsymbol{\Phi}^T \big(\boldsymbol{H}^T\big)^\star \\ &= \boldsymbol{H}^\star \boldsymbol{\Phi} \big(\boldsymbol{H}^T\big)^\star + \boldsymbol{\Psi}^\star. \end{aligned}$$

Therefore, there exist values $\hat{\boldsymbol{H}}$ and $\hat{\boldsymbol{\Psi}}$ in $\mathcal{Q}$ so that

$$\boldsymbol{S} = \hat{\boldsymbol{H}} \boldsymbol{\Phi} \hat{\boldsymbol{H}}^T + \hat{\boldsymbol{\Psi}}, \tag{I.25}$$

which was to be shown.

Appendix J

Proof of (5.28) and (5.29)

Here we wish to evaluate

$$\min_{\boldsymbol{\theta},\boldsymbol{Q}} \sum_{t=1}^{L} \int_{\chi} p(\boldsymbol{x}_k(t)|\boldsymbol{z}(t),\boldsymbol{\theta}_k)[\boldsymbol{z}(t)-\boldsymbol{\theta}\boldsymbol{\Phi}_k(t)]^H \boldsymbol{Q}^{-1}[\boldsymbol{z}(t)-\boldsymbol{\theta}\boldsymbol{\Phi}_k(t)]d\boldsymbol{x} + \ln|\boldsymbol{Q}| \quad \text{(J.26)}$$

J.1 Solution for θ_k

The problem at hand is equivalent to solving

$$\frac{\partial}{\partial \boldsymbol{\theta}} \sum_{t=1}^{L} \int_{\chi} p(\boldsymbol{x}_k(t)|\boldsymbol{z}(t),\boldsymbol{\theta}_k)[\boldsymbol{z}(t)-\boldsymbol{\theta}\boldsymbol{\Phi}_k(t)]^H \boldsymbol{Q}^{-1}[\boldsymbol{z}(t)-\boldsymbol{\theta}\boldsymbol{\Phi}_k(t)]d\boldsymbol{x} + \ln|\boldsymbol{Q}| = \boldsymbol{0}. \quad \text{(J.27)}$$

In taking the expectations, we assume $\boldsymbol{\theta}_k$ in $p(\boldsymbol{x}_k(t)|\boldsymbol{z}(t),\boldsymbol{\theta}_k)$ is held fixed at the value obtained in the previous iteration. Also, since $\boldsymbol{\theta}$ is independent of $\boldsymbol{x}$, we can move the derivative operator inside the expectation. Using the relation [77]

$$\frac{\partial}{\partial \boldsymbol{A}}(\boldsymbol{x}-\boldsymbol{A}\boldsymbol{s})^H \boldsymbol{W}(\boldsymbol{z}-\boldsymbol{A}\boldsymbol{s}) = -2\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{A}\boldsymbol{s})\boldsymbol{s}^H$$

(J.27) becomes

$$\sum_{t=1}^{L} \left\langle -2\boldsymbol{Q}^{-1}[\boldsymbol{z}(t)-\hat{\boldsymbol{\theta}}\boldsymbol{\Phi}_k(t)]\boldsymbol{\Phi}_k^H(t) \right\rangle = \boldsymbol{0}$$

where the angular brackets denote expectation w.r.t. the distribution $p(\boldsymbol{x}_k(t)|\boldsymbol{z}(t), \boldsymbol{\theta}_k)$ and $\hat{\boldsymbol{\theta}}$ is the desired estimate of $\boldsymbol{\theta}$. This is equivalent to

$$\sum_{t=1}^{L} \left\langle [\boldsymbol{z}(t) - \hat{\boldsymbol{\theta}}\boldsymbol{\Phi}_k(t)]\boldsymbol{\Phi}_k^H(t) \right\rangle = \mathbf{0}. \tag{J.28}$$

Eq. (J.28) leads to

$$\sum_{t=1}^{L} \left\langle \boldsymbol{z}(t)\boldsymbol{\Phi}_k^H(t) \right\rangle - \hat{\boldsymbol{\theta}} \sum_{t=1}^{L} \left\langle \boldsymbol{\Phi}_k(t)\boldsymbol{\Phi}_k^H(t) \right\rangle = \mathbf{0} \tag{J.29}$$

from which the result for $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_{k+1}$ follows. □

J.2 Solution for Q

The problem of relevance in this case is to solve

$$\frac{\partial}{\partial \boldsymbol{Q}} \sum_{t=1}^{L} \left\langle [\boldsymbol{z}(t) - \hat{\boldsymbol{\theta}}\boldsymbol{\Phi}_k(t)]^H \boldsymbol{Q}^{-1} [\boldsymbol{z}(t) - \hat{\boldsymbol{\theta}}\boldsymbol{\Phi}_k(t)] \right\rangle + \ln|\boldsymbol{Q}| = \mathbf{0}. \tag{J.30}$$

In this case, the distribution $p(\boldsymbol{x}_k(t)|\boldsymbol{z}(t), \boldsymbol{\theta}_k)$ is independent of $\boldsymbol{Q}$, and $\boldsymbol{Q}$ is independent of $\boldsymbol{x}$, so the derivative operation with respect to $\boldsymbol{Q}$ can be moved directly inside the expectation.

Using the following derivative rules [77]

$$\begin{aligned} \frac{\partial \boldsymbol{a}^H \boldsymbol{W}^{-1} \boldsymbol{b}}{\partial \boldsymbol{W}} &= -\boldsymbol{W}^{-H} \boldsymbol{a}\boldsymbol{b}^H \boldsymbol{W}^{-H} \\ \frac{\partial |\boldsymbol{W}|}{\partial \boldsymbol{W}} &= |\boldsymbol{W}| (\boldsymbol{W}^{-1})^H, \end{aligned}$$

(J.30) becomes

$$\begin{aligned} \sum_{t=1}^{L} \left\langle -\hat{\boldsymbol{Q}}^{-H} [\boldsymbol{z}(t) - \hat{\boldsymbol{\theta}}\boldsymbol{\Phi}_k(t)][\boldsymbol{z}(t) - \hat{\boldsymbol{\theta}}\boldsymbol{\Phi}_k(t)]^H \hat{\boldsymbol{Q}}^{-H} \right\rangle + \hat{\boldsymbol{Q}}^{-H} &= \mathbf{0} \\ \hat{\boldsymbol{Q}}^{-1} \sum_{t=1}^{L} \left\langle [\boldsymbol{z}(t) - \hat{\boldsymbol{\theta}}\boldsymbol{\Phi}_k(t)][\boldsymbol{z}(t) - \hat{\boldsymbol{\theta}}\boldsymbol{\Phi}_k(t)]^H \right\rangle &= \boldsymbol{I}_{J \times J}, \end{aligned} \tag{J.31}$$

where $\hat{\boldsymbol{Q}}$ is the desired solution. The last line follows by postmultiplication of the line above by $\hat{\boldsymbol{Q}}^H$, and recognizing that $\hat{\boldsymbol{Q}}^{-H} = \hat{\boldsymbol{Q}}^{-1}$. By substituting (J.28) into (J.31), and distributing the sum and expectation operators amongst the individual terms, we have

$$\hat{\boldsymbol{Q}} = \boldsymbol{Q}_{k+1} = \sum_{t=1}^{L} \left\langle \boldsymbol{z}(t)\boldsymbol{z}^H(t) \right\rangle - \sum_{t=1}^{L} \left\langle \hat{\boldsymbol{\theta}}\boldsymbol{\Phi}_k(t)\boldsymbol{z}^H(t) \right\rangle \tag{J.32}$$

which was to be shown. □

Bibliography

[1] R. Ahlswede and M. Burnashev. On minimax estimation in the presence of side information about remote data. *Annals of Statistics*, 18:141–171, Jan. 1990.

[2] R. Ahlswede and I. Csiszar. To get a bit of information may be as hard as to get full information. *IEEE Trans. on Information Theory*, 27:398–408, July 1981.

[3] A. Aksoy, F. Zhang, and J. W. Neilsen-Gammon. Simultaneous state and parameter estimation with an ensemble Kalman filter for thermally driven circulations. *http://www.met.tamu.edu/temp/altug_paper2_draft_report.pdf*, Aug. 2004.

[4] S. M. Alamouti. A simple transmiter diversity scheme for wireless communications. *IEEE J. Selected Areas in Communications*, 16:1451–1458, Oct 1998.

[5] C. H. Aldana, E. de Cardevalho, and J. Cioffi. Channel estimation for MISO systems using the EM algorithm. *IEEE Trans. on Signal Processing*, 51:3280–3292, 2003.

[6] S. Amari. Fisher information under restriction of Shannon information in multiterminal situations. *Ann. Inst. Statist. Math.*, 41:623–648, April 1989.

[7] S. I. Amari. Information geometry of the EM and *em* algorithm for neural networks. *Neural Networks*, 8(9):1379–1408, 1995.

[8] S. I. Amari, K. Kurata, and H. Nagaoka. Information geometry of the Boltzman machine. *IEEE Trans. on Neural Networks*, 3(2):260–272, Mar 1992.

[9] S. I. Amari and H. Nagaoka. *Methods of information geometry.* Oxford Press, 2000.

[10] S. Arulampalam and *et. al.* A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. on Signal Processing*, 50:174–188, Feb. 2002.

[11] A. Bagchi and P. ten Brummelhuis. Simultaneous ML estimation of state and parameters for hyperbolic systems with noisy boundary conditions. In *Proc. of the 29th. Conference on Decision and Control*, Hawaii, 1990.

[12] J. Bajcsy and P. Mitran. Coding for the Slepian-Wolf problem with turbo codes. In *Proc. of the Global Telecommunications Conference*, San Antonio, Texas, U.S.A, 2001.

[13] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Application to Tracking and Navigation.* John Wiley and Sons, Inc., New York, 2001.

[14] O. Barndorff-Neilsen. *Information and exponential families in statistical theory.* Wiley Interscience, Chichester, 1978.

[15] T. Berger. Decentralized estimation and decision theory. In *Proc. of the IEEE 7th Spring Workshop on Information Theory*, Mt. Kisco, NY, U.S.A, 1979.

[16] N. Bergman. *Recursive Bayesian Estimation: Navigation and Tracking Applications.* PhD thesis, May 1999.

[17] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[18] C. Byrne. Iterative image reconstruction algorithms based on cross-entropy minimization. *IEEE Trans. on Image Processing*, 2:96–103, 1993.

[19] W. J. Byrne. Alternating minimization and Boltzman machine learning. *IEEE Trans. on Neural Networks*, 3(4):612–620, Apr 1992.

[20] W. J. Byrne. Information geometry and maximum likelihood criteria. In *Conference on Information Sciences and Systems*, Princeton, NJ, 1996.

[21] J. F. Cardoso and A. Souloumiac. Blind beamforming for non-Gaussian signals. *IEE-Proceedings-F*, 140(6):362–370, 1993.

[22] T. P. Coleman and *et al.* Low-complexity approaches to SlepianWolf near-lossless distributed data compression. *IEEE I-IT*, 52:3546–3561, Aug. 2006.

[23] T. M. Cover, A. E. Gamal, and M. Salehi. Multiple access channels with arbitrarily correlated sources. *IEEE Trans. on Information Theory*, 26:648–657, Nov. 1980.

[24] T. M. Cover and J. Thomas. *Elements of information theory.* Wiley Interscience, New York, 1991.

[25] H. Cox. On the estimation of state variables and parameters for noisy dynamic systems. *IEEE Trans. on Automatic Control*, 9:5–12, Jan. 1964.

[26] C. Cozzo and B. L. Hughes. Joint channel estimation and data detection in space-time communications. *IEEE Trans. on Communications*, 51(8):1266–1270, Aug. 2003.

[27] I. Csiszar. I-divergence, geometry of probability distributions and minimization problems. *The Annals of Probability*, 3(1):146–158, 1975.

[28] I. Csiszar. Information theoretic method in probability and statistics. In *http://ieeeits.org/publications/nltr/98_mar/01csi.pdf*, 1998.

[29] I. Csiszar. Method of types. *IEEE Trans. on Information Theory*, 44:2505–2523, Oct. 1998.

[30] I. Csiszar and J. Korner. *Information Theory: Coding Thoerems for Discrete Memoryless Systems.* Academic Press, New York, 1981.

[31] I. Csiszar and P. Shields. Information theory and statistics: a tutorial. In *http://www.math.utoledo.edu/ pshields/latex.html*, 2005.

[32] I. Csiszar and G. Tusnady. Information geometry and alternating minimization procedures. *Statistics and Decision*, (1):205–237, 1984.

[33] J. Dauwels, S. Korl, and H. A. Loeliger. Expectation maximization as message passing. In *Proc. of the IEEE ISIT-2005*, pages 498–519, Sept. 2005.

[34] M. H. DeGroot. *Probability and Statistics.* Addison–Wesley Publications, 1995.

[35] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data. *Journal of the Royal Statistical Society*, 39:1–38, Series B 1967.

[36] A. Doucet, N. de Freitas, and N. G. Eds. *Sequential Monte Carlo Methods in Practice.* Springer-Verlag, New York, 2001.

[37] A. Doucet, S. Godsill, and C. Andrieu. On Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, pages 197–208, Oct. 2000.

[38] A. W. Eckford. Channel estimation in block fading channels using the factor graph EM algorithm. In *Proc.* 22^{nd} *Biennial Symposium on Communications*, pages 498–519, Kingston, Ontario, Canada, May 31-June 3 2004.

[39] A. W. Eckford. The factor graph EM algorithm: Applications for LDPC codes. In *Proc. 6th IEEE Workshop on Signal Processing Advances for Wireless Communication (SPAWC)*, New York, USA, 2005.

[40] M. Feder and J. A. Capitovic. Alorithm for joint channel estimation and data recovery- application tro equalization in underwater communications. *IEEE Trans. on Oceanic Engineering*, 16(1):42–55, 1991.

[41] J. A. Fessler and A. O. Hero. Space-alternating generalized expectation maximization algorithm. *IEEE Trans. on Signal Processing*, 42(10):2664–2667, Oct. 1994.

[42] R. G. Gallager. *Low Density parity Check Codes.* MIT Press, Cambridge, MA, 1963.

[43] A. E. Gamal. A simple proof of the Ahlswede-Csiszar one-bit theorem. *IEEE Trans. on Information Theory*, 29:931–933, Nov. 1983.

[44] M. Gastpar. *To code or not to code, Ph.D. Thesis.* EPFL, Switzerland, 2003.

[45] N. Gehrig and P. L. Dragotti. Symmetric and a-symmetric Slepian-Wolf codes with systematic and non-systematic linear codes. *IEEE Communications Letters*, 9:61–63, Jan. 2005.

[46] G. H. Golub and C. F. V. Loan. *MATRIX Computations.* The Johns Hopkins University Press, Baltimore, MA, 2nd edition, 1993.

[47] T. S. Han and S. Amari. Parameter estimation with multiterminal data compression. *IEEE Trans. on Information Theory*, 41:1802–1833, Nov. 1995.

[48] T. S. Han and S. Amari. Statistical inference under multiterminal data compression. *IEEE Trans. on Information Theory*, 44:2300–2324, Oct. 1998.

[49] V. Havlena. Simultaneous parameter tracking and state estimation in a linear systems. *Automatica*, 29:1041–1052, July 1993.

[50] V. Havlena, J. Stecha, and T. Pajdia. Smoothing preserving discontinuity based on alternative models of parameter development. In *Proceednigs of the Czech Pattern Recognition Workshop*, pages 75–83, Prague, Czechoslovak, 1993.

[51] S. Haykin. *Communication Systems, 4th Edition.* John Wiley and Sons Inc., New York, 2001.

[52] C. G. Hilborn and D. G. Lainiotis. Optimal estimation in the presence of unknown parameters. *IEEE Trans. on Systems, Science, and Cybernetics*, 5:38–43, Jan. 1969.

[53] Y. Ho and B. Whalen. An approach to the identification and control of linear dynamics systems with unknown parameters. *IEEE Trans. on Automatic Control*, 8:255–256, July 1963.

[54] R. A. Iltis and S. Kim. Geometric derivation of EM and generelizaed succesive interference cancellation algorithms with applications to CDMA channel estimation. *IEEE Trans. on Signal Processing*, 51(5):1367–1377, May 2003.

[55] R. Jornsten. *Data compression and Its Statistical Implications, with an Application to the Analysis of Microarray Images, Ph.D. Thesis.* University of Berkeley, 2001.

[56] S. Julier and J. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92:401–422, March 2004.

[57] G. K. Kaleh and R. Vallet. Joint parameter estimation and symbol detection for linear and nonlinear unknown channels. *IEEE Trans. on Communications*, 42:2406–2413, July 1994.

[58] R. E. Kass and P. W. Vos. *Geometrical foundations of asymptotic inference.* Wiley Interscience, New York, 1997.

[59] G. Kitagawa. Monte calro filter and smoother for non-Gaussian nonlinear state space models. *J. Computational and Graphical Statistics*, 5:1–25, Jan. 1996.

[60] A. Kocian and B. H. Fleury. EM-based joint data detection and channel estimation of dc-cdma signals,. *IEEE Trans. on Communications*, 51(10):1709–1720, Oct. 2003.

[61] J. Korner and J. Marton. How to encode the modulo-two sum of binary sources. *IEEE Trans. on Information Theory*, 25:219–221, March 1979.

[62] F. R. Kschischang, B. J. Frey, and H. A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. on Information Theory*, 4:498–519, Feb. 2001.

[63] X. R. Li and Y. Zhang. Multiple-model estimation with variable structure. *IEEE Trans. on Automatic Control*, 41:478–493, April 1996.

[64] X. R. Li and Y. Zhang. Multiple-model estimation with variable stucture, likely-model set algorithm. *IEEE Trans. on Aerospace and Electronic Systems*, 36:448–466, April 2000.

[65] X. Lin, T. Kirubarajan, Y. Bar-Shalom, and S. Maskell. Comparison of EKF, pseudomeasurement and particle filters for bearing-only target tracking problem. In *Proc. SPIE Conference on Image and Signal Processing for Small Targets*, Orlando, FL, 2002.

[66] A. Liveris, Z. Xiong, and C. Georghiades. Compression of binary sources with side information at the decoder using LDPC codes. *IEEE Communicatins Letters*, 6:440–442, Oct. 2002.

[67] L. Ljung and T. Soderstorm. *Theory and Practice of Recursive Identification.* MIT Press, Cambridge, MA, 1983.

[68] X. Ma, H. Kobayashi, and S. C. Schwartz. An enhanced channel estimation algorithm for OFDM: Combined EM algorithm and polynomial fitting. In *Prof. of the IEEE ICASSP 2003*, Hong Kong, 2003.

[69] D. T. Magill. Optimal adaptive estimation of sampled stochastic processes. *IEEE Trans. on Automatic Control*, pages 434–439, Oct 1965.

[70] J. H. Manton and Y. Hua. Maximum-likelihood algorithms for deterministic and semi-blind channel identification. In *Second International Conference on Information, Communications and Signal Processing*, Singapore, 1999.

[71] L. Mazet and *et. al.* EM-based semi-blind estimation of time-varying channels. *IEEE Trans. on Signal Processing*, 52(2):406–417, feb 2004.

[72] E. Mazor and *et. al.* Interacting multiple model methods in target tracking: A survey. *IEEE Trans. on Aerospace and Electronic Systems*, 34:103123, Jan. 1996.

[73] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions.* John Wiley and Sons Inc., New York, 1997.

[74] P. Mitran and J. Bajcsy. Near shannon-limit coding for the Slepian-Wolf problem. In *Proc. of the 21st Biennial Symposium on Communications*, Kingston, Ontario, Canada, 2005.

[75] P. Mookerjee and F. Reifler. Reduced state estimator for systems with parametric inputs. *IEEE Trans. on Aerospace and Electronic Systems*, 40:446–461, April 2004.

[76] L. W. Nelson and E. Stear. The simultaneus on-line estimation of parameters and states in linear systems. *IEEE Trans. on Automatic Control*, 21:94–98, Feb. 1976.

[77] K. B. Petersen and M. S. Pedersen. The Matrix Cookbook, http://www2.imm.dtu.dk/pubdb/, Feb 2007.

[78] S. Pradhan and K. Ramchandran. Distributed source coding: Symmetric rates and applications to sensor networks. In *Proc. of the Conference on Data Compression*, Salt Lake City, Utah, U.S.A, 2000.

[79] S. Pradhan and K. Ramchandran. Distributed source coding with syndromes (DISCUS): Design and construction. *IEEE Trans. on Information Theory*, 49:626–643, March 2003.

[80] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Trans. on Information Theory*, 47:619–637, Feb. 2001.

[81] B. Ristic and N. Okello. Sensor registration in ECEF coordinates using the MLR algorithm. In *Proceedings of the 6th International Conference on Information Fusion*, Cairns, Australia, July 2003.

[82] J. Roweis and Z. Ghahramani. *Learning Nonlinear Dynamical Systems using the Expectation-Maximization Algorithm, in S. Haykin, Ed., Kalman Filtering and Neural Networks.* Dover Publications, New York, 2001.

[83] I. Runsak. Multiple objective optimization approach to simultaneuos identification and tracking of uncertain systems. In *Proc. of 10th. IFAC Workshop on Control Applications of Optimization*, Haifa, Israel, Dec. 19-21, 1995.

[84] W. E. Ryan. *An Introduction to LDPC codes.* CRC Handbook for Coding and Signal Processing for Recoding Systems, B. Vasic, ed., CRC Press,, 2004.

[85] K. R. S. S. Pradhan. Distributed source coding using syndromes (DISCUS): Design and construction. In *Proc. of the Conference on Data Compression*, Salt Lake City, Utah, U.S.A, 1999.

[86] D. Schonberg, K. Ramchandran, and S. Pradhan. Distributed code constructions for the entire Slepian-Wolf rate region for arbitrarily correlated sources. In *Proc. of the Conference on Data Compression*, Salt Lake City, Utah, U.S.A, 2004.

[87] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, July and Oct. 1948.

[88] H. Shimokawa and S. I. Amari. Multiterminal estimation theory with binary symmetric source. In *Proc. of the IEEE ISIT-1995*, page 447, 1995.

[89] M. A. Shokrollahi. Ldpc codes: An introduction. *"http://algo.epfl.ch/contents/output/pubs/ldpc_intro.pdf"*, 2002.

[90] D. Slepian and J. Wolf. Noiseless coding of correlated information sources. *IEEE Trans. on Information Theory*, 19:471–480, July 1973.

[91] V. Stankovic and *et. al.* On code design for the SlepianWolf problem and lossless multiterminal networks. *IEEE Trans. on Information Theory*, 52:1495–1507, April 2006.

[92] G. Taricco and E. Biglieri. Space-time decoding with imperfect channel estimation. *IEEE Trans. on Wireless Communications*, 4:1874–1888, July 2005.

[93] V. Tarokh, N. Seshardi, and A. R. Calderbank. Space-time codes for high data rate wireless communication: Performance criteria and code construction. *IEEE Trans. on Information Theory*, 44:744–765, 1998.

[94] I. E. Telatar. Capacity of multi-antenna Gaussian channels. *AT&T Bell Labs Internal Tech. Memo.*, 1995.

[95] P. Tichavsky, C. Muravchik, and A. Nehorai. Posterior Cramer-Rao bounds for discrete–time nonlinear filtering. *IEEE Trans. on Signal Procesing*, 46:1386–1396, May 1999.

[96] L. Tong and S. Perreau. Multichannel blind identification: from subspace to maximum likelihood methods. *Proceedings of the IEEE*, 86:1951–1968, Oct. 1998.

[97] J. K. Tugnait. Adaptive estimation and identification for discrete systems with Markov jump parameters. *IEEE Trans. on Automatic Control*, 27:1054106, Oct 1982.

[98] J. K. Tugnait and A. H. Haddad. Adaptive estimation in linear systems with unknown Markovian noise statistics. *IEEE Trans. on Information Theory*, 26:6678, Jan. 1980.

[99] J. K. Tugnait, L. Tong, and Z. Ding. Single–user channel estimation and equalization. *IEEE Signal Processing Magazine*, 17:17–28, May 2000.

[100] C. F. J. Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, March 1983.

[101] C. F. J. Wu. On the convergence properties of the em algorithm. *Annals of Statistics*, 11:95–103, 1983.

[102] A. Wyner. Recent results in the shannon theory. *IEEE Trans. on Information Theory*, 20:2–10, Jan. 1974.

[103] J. J. Xiao and Z. Q. Luo. Multiterminal source-channel communication under orthogonal multiple access. *"http://www.ece.umn.edu/users/xiao/pubs/sc_multiterminal.pdf"*, 2005.

[104] Z. Xiong, A. Liveris, and S. Cheng. Distributed source coding for sensor networks. *IEEE Signal Processing Magazine*, pages 80–94, Sept. 2004.

[105] H. Zamiri-Jafarian and S. Pasupathy. Recursive channel estimation for wireless communication via the EM algorithm geometric derivation of EM and generalized succesive interference cancellation algorithm,. In *ICPWC 1997*, 1997.

[106] Z. Zhang and T. Berger. Estimation via compressed information. *IEEE Trans. on Information Theory*, 34:198–211, March 1988.

[107] A. Zia, J. P. Reilly, T. Kirubarajan, and S. Shirani. An EM algorithm for nonlinear state estimation with model uncertainties. *Accepted to publish in IEEE Trans. on Signal Processing.*

[108] A. Zia, J. P. Reilly, T. Kirubarajan, and S. Shirani. A stochastic EM algorithm for nonlinear state estimation with model uncertainties. In *Proc. of SPIE Annual Meeting Symposium*, San Diego, U.S.A., Aug. 2003.

[109] A. Zia, J. P. Reilly, J. Manton, and S. Shirani. An information geometric approach to ml estimation with incomplete data: application to semi-blind MIMO channel identification. *Accepted for publication in IEEE Trans. on Signal Processing.*

[110] A. Zia, J. P. Reilly, and S. Shirani. Distributed estimation; theorems for sufficient and necessary rates. *Prepared for submission to IEEE Trans. on Information Theory.*

[111] A. Zia, J. P. Reilly, and S. Shirani. Distributed parameter estimation: a factor graph approach. *Prepared for submission to IEEE Trans. on Communications.*

[112] A. Zia, J. P. Reilly, and S. Shirani. An information geometric approach to channel identification. In *Proc. of the IEEE ICASSP'04*, Montreal, Canada, May 2004.

[113] A. Zia, J. P. Reilly, and S. Shirani. Information geometric approach to channel identification: a comparison with EM-MCMC algorithm. In *Proc. of the IEEE ICC'04*, Paris, France, June 2004.

[114] A. Zia, J. P. Reilly, and S. Shirani. Distributed estimation; three theorems. In *Proc. of the IEEE ITW'07*, Lake Tahoe, CA, U.S.A, Sept. 2007.

[115] A. Zia, J. P. Reilly, and S. Shirani. Distributed parameter estimation with side–information: a factor graph approach. In *Proc. of the IEEE ISIT'07*, Nice, France, June 2007.